

A vertical decorative bar on the left side of the page, filled with various black geometric shapes including circles, squares, triangles, and lines of different sizes and orientations.

**LAMK**

Lahden ammattikorkeakoulu  
Lahti University of Applied Sciences

# MOBILE-FIRST JA MATERIAL DESIGN KÄYTTÖLIITTYMÄ- SUUNNITTELUPROSESSISSA

Case: Salpausselkä Geopark -mobiiliapplikaatio

LAHDEN AMMATTIKORKEAKOULU  
Tieto- ja viestintätekniikka  
Mediatekniikka  
Opinnäytetyö  
Kevät 2018  
Juha Laurikainen

## Tiivistelmä

Tekijä Laurikainen, Juha	Julkaisun laji Opinnäytetyö, AMK	Valmistumisaika Kevät 2018
	Sivumäärä 29, 9 liitesivua	
Työn nimi <b>MOBILE-FIRST JA MATERIAL DESIGN KÄYTTÖLIITTYMÄSUUNNITTELUPROSESSISSA</b> Case: Salpausselkä Geopark -mobiiliapplikaatio		
Tutkinto Tieto- ja viestintätekniikan koulutusohjelma, Mediatekniikka		
<b>Tiivistelmä</b> <p>Opinnäytetyön tavoite oli suunnitella käyttöliittymä Salpausselkä Geopark -mobiiliapplikaatiolle. Tämän applikaation tulisi tarjota käyttäjälle tietoa Geoparkin sisältämistä kohteista ja toimia navigaatiotyökaluna Geoparkin alueella liikuttaessa.</p> <p>Suunnittelua lähestyttiin käyttäen mobile-first -näkökulmaa. Käyttöliittymän ulkoasun toteutukseen valittiin Material Design -suunnitteluohjeisto, jonka tarjoamilla säännöillä ja niiden pohjalta tehdyillä työkaluilla suunnitteluprosessia voitiin helpottaa. Suunnittelun iteroinnin aikana toteutettiin käytettävyydestä käyttäen Proto.io-prototypointiverkkopalvelua.</p> <p>Työn tulokseksi saatiin älypuhelinkoon käyttöliittymäsuunnitelma, jota voidaan jatkossa hyödyntää mobiiliapplikaation toteutuksessa. Lisäksi otettiin myös kantaa siihen, miltä tablettikoon käyttöliittymän tulisi näyttää.</p>		
<b>Avainsanat</b> heuristiikka, käytettävyys, käyttäjäkokemus, käyttöliittymäsuunnittelu, käyttöliittymä, Material Design, mobile-first, prototypointi, rautalankamalli		

## Abstract

Author Laurikainen, Juha	Type of publication Bachelor's thesis	Published Spring 2018
	Number of pages 29, 9 appendices	
Title of publication <b>User interface design using mobile-first approach and Material Design</b> CASE: Salpausselkä Geopark mobile application		
Name of Degree Degree Programme in Information and Communications Technology		
<p>Abstract</p> <p>The goal of this thesis was to design a user interface for the Salpausselkä Geopark mobile application. This application should provide the user with information about the different sites within the Geopark and work as a navigation tool when the user is visiting the Geopark.</p> <p>The design process was approached using mobile-first principles. Material Design documentation was used to define the look of the application. The design guidelines and available tools for Material Design made the design process easier. During the design iteration, usability testing was performed using prototypes of the interface made with the Proto.io prototyping platform.</p> <p>As a result, a smartphone-sized design for the application's user interface was produced. This design can be utilized when developing the application in the future. The design was also expanded to give an idea of how the interface should look on a tablet.</p>		
<p>Keywords</p> <p>heuristics, usability, user experience, user interface design, user interface, Material Design, mobile-first, prototype, wireframe</p>		

## SISÄLLYS

1	JOHDANTO .....	1
2	MOBILE-FIRST .....	2
2.1	Miksi mobile-first? .....	2
2.2	Graceful degradation ja progressive enhancement .....	3
2.3	Mobiiliapplikaatiotyypit .....	4
2.3.1	Natiiviapplikaatiot .....	4
2.3.2	Web-applikaatiot .....	4
2.3.3	Hybridiapplikaatiot .....	6
2.3.4	Applikaatiotyypin valinta .....	6
3	KÄYTTÖLIITYMÄ JA KÄYTTÄJÄKOKEMUS .....	7
3.1	Käytettävyys .....	8
3.2	Heuristiikka .....	8
4	KÄYTTÖLIITTYMÄSUUNNITELUN TYÖNKULKU .....	12
4.1	Rautalankamallinnus .....	12
4.1.1	Käytön kartoitus .....	12
4.1.2	Visuaalinen rautalankamallinnus .....	13
4.2	Korkeatarkkuuksinen ulkoasumallinnus ja prototypointi .....	13
5	MATERIAL DESIGN -SUUNNITTELUOHJEISTO .....	16
5.1	Material-tila .....	17
5.2	Material-kappaleet .....	18
5.3	Material-liike .....	19
5.4	Material-työkalut .....	20
6	CASE: SALPAUSSELKÄ GEOPARK -MOBIILIIAPPLIKAATIO .....	21
6.1	Lähtötilanne .....	21
6.2	Rautalankamallin toteutus .....	22
6.3	Korkeatarkkuuksinen ulkoasu .....	23
6.4	Prototypointi ja käytettävyytestaus .....	25
7	YHTEENVETO .....	29
	LÄHTEET .....	30
	LIITTEET .....	33

## 1 JOHDANTO

Mobile-first on suunnittelun lähestymistapa, jonka tarkoitus on taata mahdollisimman hyvä käyttäjäkokemus mobiilikäyttäjälle. Tämä lähestymistapa on tärkeä web- ja applikaatio-suunnitteluun, koska jo vuosia mobiililaitteiden käyttäjien määrä on ollut kasvussa ja on saavuttanut määrällään työpöytäkäyttäjät. Myös mobiiliapplikaatioiden tulisi olla skaalautuvia, koska mobiilipäätelaitteita on mittavissa määrin eri resoluutioita ja pikselitarkkuuksia. Tämä tarkoittaa sitä, että yhdelle laitteelle suunniteltu käyttöliittymä ei pysty tuottamaan kattavaa käyttökokemusta kaikille päätelaitteille.

Mobile-first -ideologian pohjalta on toteutettu useita webkehyskiä, joilla voidaan vaivattomasti luoda responsiivisia websivuja. Näitä webkehyskiä voidaan käyttää hyväksi myös mobiiliapplikaatioita luodessa, kun niitä tuotetaan hybridiapplikaatioina. Esimerkiksi Googlen omille verkkopalveluille ja laitteita kehittämän Material Design -ohjeiston pohjalta on toteutettu useita tällaisia webkehyskiä.

Tässä opinnäytetyössä lähdettiin toteuttamaan käyttöliittymäsuunnitelmaa Salpausselkä Geoparkille. Salpausselkä Geopark -projekti on Lahden ammattikorkeakoulun hallinnoima hanke, jonka tarkoitus on liittää Salpausselkä UNESCO Global Geopark -verkostoon. Yhtenä Salpausselkä Geopark -hankkeeseen liittyvänä tavoitteena on tuottaa Salpausselkä Geoparkille mobiiliapplikaatio, jolla voidaan tarjota tietoa käyttäjälle Geoparkin sisältämistä geologisesti merkittävistä kohteista. Mobiiliapplikaation tulisi toimia myös navigaatio-työkaluna käyttäjän ollessa liikkeellä Geopark-alueella.

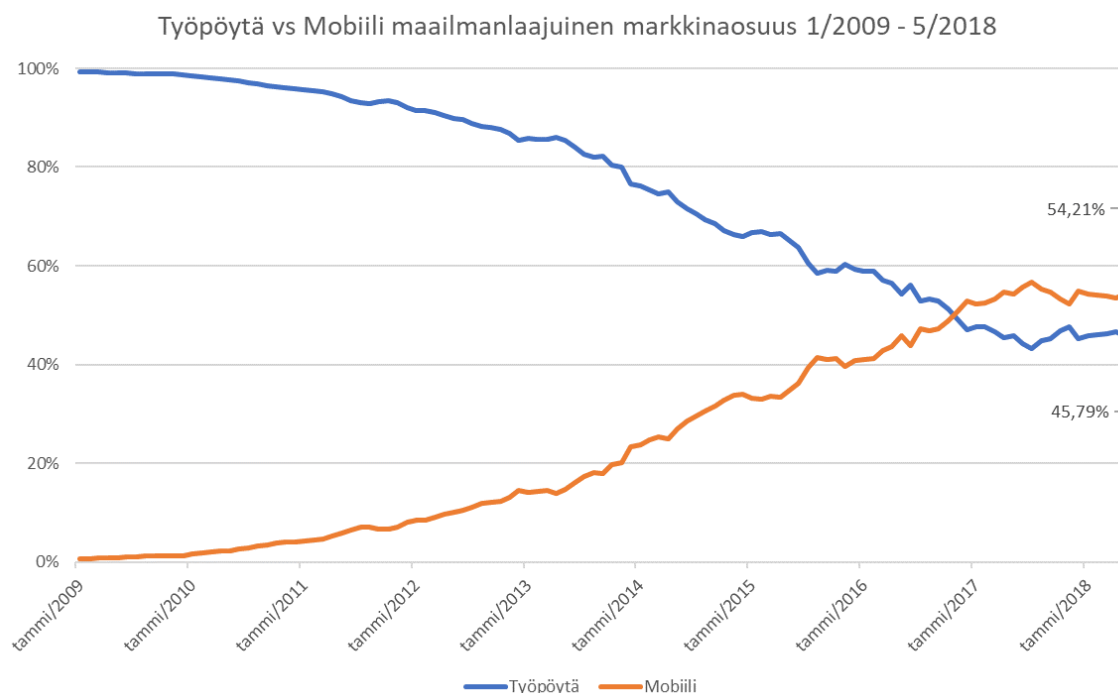
Mobiiliapplikaatio tultaisiin toteuttamaan hybridiapplikaationa, webteknologioita käyttäen, jotta tarpeen vaatiessa se voitaisiin mukauttaa vähin vaivoin myös työpöytäselainkäyttöön. Opinnäytetyön tavoite oli suunnitella käyttöliittymä tälle mobiiliapplikaatiolle, mobile-first- ja Material Design -suunnitteluperiaatteita käyttäen.

## 2 MOBILE-FIRST

### 2.1 Miksi mobile-first?

Mobile-first tarkoittaa suunnittelun lähestymistapaa, jossa suunnittelu alkaa pienimmistä päätelaitteista aloittaen. Suunnittelua jatketaan tämän jälkeen suuremmille päätelaitteille, lisäten sisältöä ja toiminnallisuutta sen ollessa mahdollista. Tätä lähestymistapaa kutsutaan myös nimellä progressive enhancement, ja sen vastakohtainen suunnittelun lähestymistapa on graceful degradation. (Gremillion 2018.)

Mobile-first-termiä on ensimmäiseksi käyttänyt Luke Wroblewski blogikirjoituksessaan webapplikaatioiden suunnittelusta, vuonna 2009. Blogissaan hän määrittää syitä, miksi mobile-first-lähestymistapa suunnitteluun on tärkeää ja rajaa tälle kolme pääsyytä. Ensimmäinen näistä syistä oli mobiililaitteiden räjähdysmäinen kasvu, joka on nähtävillä päätelaitteiden maailmanlaajuisia markkinaosuuksia kuvaavassa kuviossa 1. Toinen syy oli mobiilin vaikutus sisällön rajaamiseen, mikä tarkoittaa, että esitystilan ollessa rajattu on pakko keskittyä oleellisimpaan sisältöön ja toimintoihin. Viimeiseksi syyksi Wroblewski oli antanut mobiililaitteiden uudet laitteisto-ominaisuudet, jotka ovat ohittamassa loppuun kehitetyt front-end -teknologiat vuorovaikutuksen näkökulmasta. (Wroblewski 2009; Wroblewski 2010.)



Kuvio 1. Mobiililaitteiden ja työpöytätietokoneiden markkinaosuus, 1/2009 – 5/2018 (Statcounter 2018)

## 2.2 Graceful degradation ja progressive enhancement

Suunnitteluprosessia voidaan lähestyä kahdella eri tavalla, molemmilla tavoilla pyritään aikaansaamaan tuote, joka olisi kaikkien käytettävissä. Näiden tapojen nimet ovat graceful degradation ja progressive enhancement. (W3C 2015.)

Graceful degradation on lähestymistapa suunnitteluun, jolla pyritään tuottamaan tietty käyttökokemus nykyaikaisille selaimille tai päätelaitteille. Tarpeen vaatiessa, tästä käyttökokemuksesta tingitään. Tinkiminen tapahtuu vanhempien selaimien tai päätelaitteiden teknologioiden ollessa riittämättömiä tuottamaan haluttua lopputulosta. Tästä seuraa, että osa käyttäjien käyttökokemuksista kärsii, mutta voidaan silti tuottaa toimiva käyttökokemus tuotteen perustoimintojen rikkoutumatta. (W3C 2015.)

Graceful degradationin päinvastaista tapaa lähestyä suunnittelua kutsutaan nimellä Progressive enhancement. Tällä suunnittelutavalla pyritään tuottamaan kaikki selaimet tai päätelaitteet kattava peruskäyttökokemus, jolle myös toteutetaan kehittyneempiä toiminnollisuuksia, niiden ollessa käytettävissä. (W3C 2015.)

Progressive enhancementia pidetään yleisesti parempana suunnittelutapana näistä kahdesta lähestymistavasta, koska voidaan tuottaa tuote, joka toimii toimintaympäristöstä riippumatta ja jonka päälle voidaan uusien teknologioiden tullessa laajemmin käyttöön lisätä

toiminnallisuutta. Progressive enhancement -suunnittelutavalla voidaan taata myös vaakaammin käytettävä tuote, vaikkakin tämä lähestymistapa vaatii alustavasti enemmän työtä. Graceful degradation suunnittelutapaa on taas helpompi käyttää esimerkiksi olemassa olevan tuotteen korjaamiseen, ja se vaatii suunnittelun puolesta vähemmän työtä ollen kuitenkin tuotteen koon laajetessa vaikeammin hallittavissa. (W3C 2015.)

## 2.3 Mobiiliapplikaatiotyypit

Mobiiliapplikaatiot voidaan jakaa natiivi-, web- ja hybridiapplikaatioihin päätelaitteesta riippumatta, kunhan laitteeseen on mahdollista vain asentaa ohjelmia ja sillä on pääsy internetiin. Näistä applikaatiotyypeistä natiivi- ja hybridiapplikaatiot toimivat hyvin saman tapaisesti: molemmat niistä pakataan jaettavaksi päätelaitteiden omaan markkinapaikkaan tai vastaavaan jakelukanavaan, joista ne ovat asennettavissa. Web-applikaatiot ovat vain päätelaitteisiin kirjanmerkeiksi lisättyjä websivuja, jotka sitten pystyvät toimimaan muiden applikaatioiden tapaan kokoruututilassa. (Banga & Weinhold 2014, 82 - 85.)

### 2.3.1 Natiiviapplikaatiot

Natiiviapplikaatiot ovat ohjelmia, jotka on kokonaisuudessaan kirjoitettu päätelaitteen omaa ohjelmointikieltä käyttäen ja seuraten laitteen valmistajan määrittämän ohjeiston parhaita käytäntöjä. Käyttöliittymän toteuttamisessa natiiviapplikaatiot voivat hyödyntää laitteen käyttöjärjestelmään sisäänrakennettuja sovelluskehyskomponentteja. (Banga & Weinhold 2014, 82 - 83.)

Natiiviapplikaatioilla on pääsy laitteen kaikkiin laitteisto-ominaisuuksiin, kuten esimerkiksi kameraan ja GPS:ään, lisäksi tämän tyyppiset applikaatiot voivat hyödyntää laitteen eleohjausta (Budi 2013). Natiiviapplikaatiot ovat myös yleisesti ottaen suorituskyvyltään tehokkain applikaatiotyyppi, koska ne pystyvät hyödyntämään laitteiston resursseja parhaiten esimerkiksi 3D-kiihdytykseen ja animaatioihin. (Banga & Weinhold 2014, 82 - 83.)

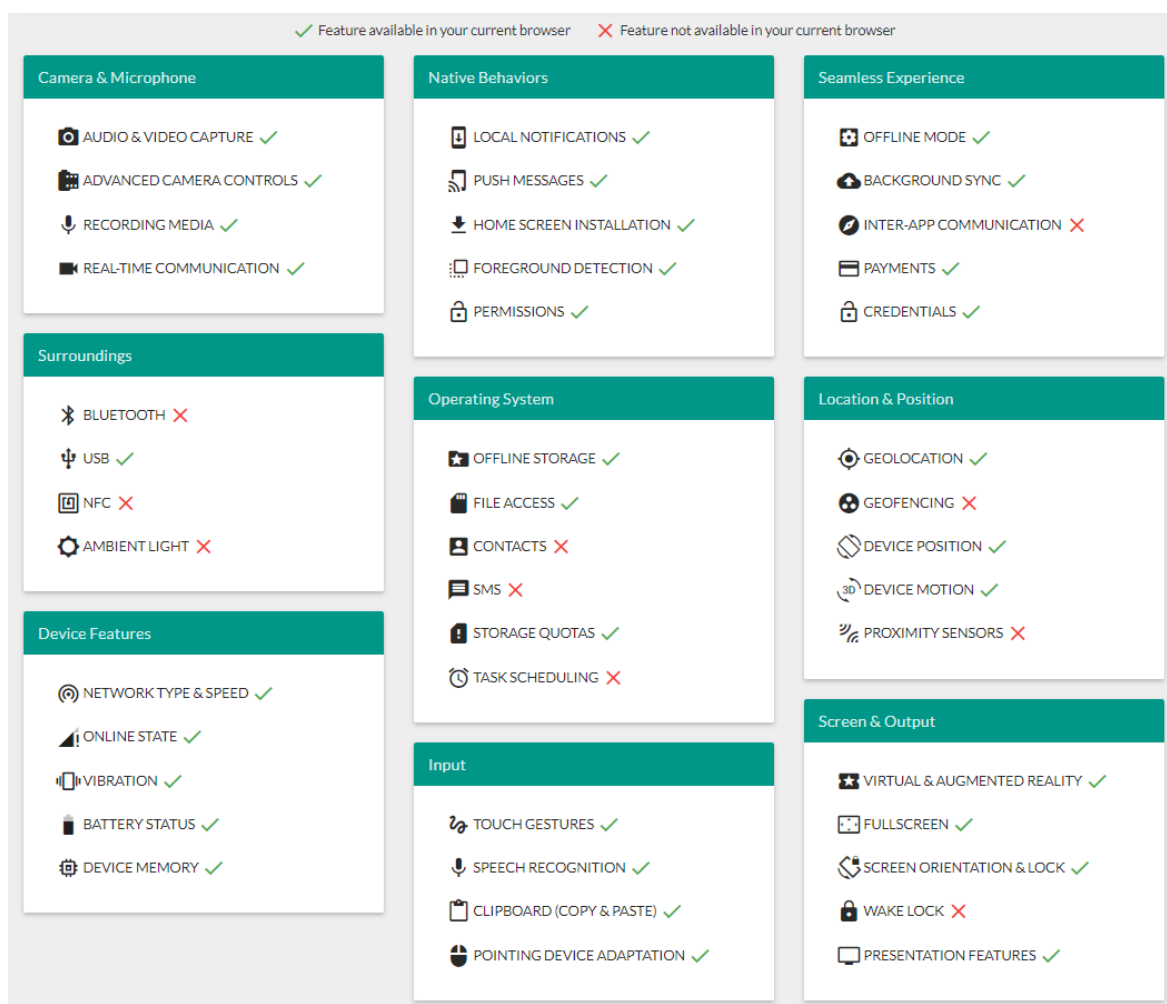
### 2.3.2 Web-applikaatiot

Web-applikaatiot ovat natiiviapplikaatioiden vastakohta; ne toimivat kokonaisuudessaan päätelaitteen selaimessa, websivun tapaan, käyttäen hyväkseen webteknologioita, kuten HTML5, CSS ja JavaScript. Web-applikaatiot tarvitsevat toimiakseen vain modernin, webstandardeja tukevan selaimen. (Banga & Weinhold 2014, 83 - 84.)

Selaimessa toimiva webapplikaatio ei voi hyödyntää päätelaitteen kaikkia laitteisto- tai käyttöjärjestelmäominaisuuksia (Budi 2013). HTML5-ominaisuudet ovat kuitenkin kehitty-



neet nykyään siten, että suurin osa laitteisto- ja käyttöjärjestelmäominaisuuksista on käytettävissä päätelaitteesta ja sen selaimesta riippuen, kuten kuvion 2 esittämästä tuettujen HTML5-ominaisuuksien ruutukaappauksesta Chrome-selaimen versiossa 65.0.3325.181 voidaan päätellä. (Bar 2018, 83 - 84.)



Kuvio 2. Chrome-selaimen tukemat HTML5-ominaisuudet (Bar 2018)

Käyttöliittymän osalta webapplikaatiot eivät voi suoraan hyödyntää päätelaitteen valmiita sovelluskehyskomponentteja (Banga & Weinhold 2014). Web-applikaation käyttöliittymä on kuitenkin mahdollista tyyliellä päätelaitteen natiivityyliseksi käyttäen tähän tarkoitukseen luotuja CSS-tyylikirjastoja (Kharlampidi 2016; Materialize 2018).

### 2.3.3 Hybridiapplikaatiot

Hybridiapplikaatiot yhdistävät natiivi- ja webapplikaation ominaisuuksia. Tämä applikaatio-tyyppi toimii laitteen selainta hyödyntäen, ja tämä selain on pakattu ”natiiviapplikaatio-kuoren” sisälle, antaen applikaatiolle webapplikaatiota laajemman pääsyn päätelaitteen laitteisto-ominaisuuksiin. (Budiou 2013; Banga & Weinhold 2014, 84 - 85.)

Hybridiapplikaatio pystyy hyödyntämään uudelleen käytettäviä, alustariippumattomia komponentteja, jotka on tuotettu webteknologioin. Lisäksi hybridiapplikaatioille voidaan tuottaa tarvittaessa alustariippuvaisia ominaisuuksia natiiviteknologioin. (Banga & Weinhold 2014, 84-85.)

### 2.3.4 Applikaatiotyyppin valinta

Kaikilla edellä mainituilla applikaatiotyypeillä voidaan täyttää käyttäjän tarpeita, ja yksikään näistä applikaatiotyypeistä ei ole universaalisti paras ratkaisu jokaiseen tilanteeseen. Applikaatiotyyppin valinnassa tulisi ottaa huomioon applikaatiota työstävän osapuolen tarpeet, ja myös eri applikaatiotyyppien vahvuudet ja heikkoudet. (Budiou 2013; Marketing & Growth Hacking 2017.)

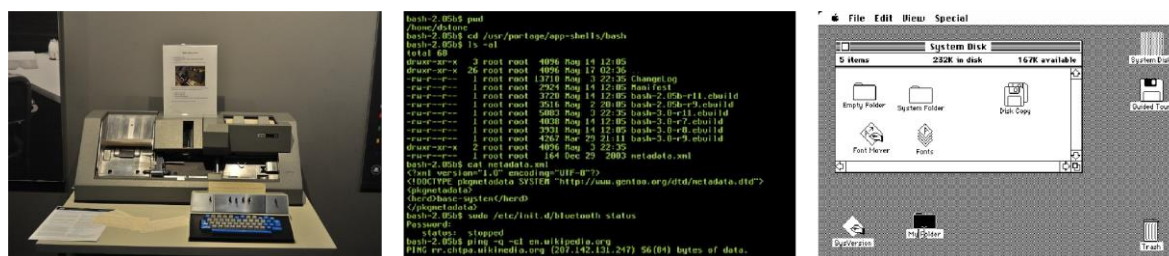
Natiiviapplikaatioilla on paras suorituskyky ja pääsy laitteen kaikkiin ominaisuuksiin. Ne toimivat myös parhaiten offline-tilassa ja huonoilla yhteyksillä, tosin web- ja hybridiapplikaatioissa voidaan hyödyntää HTML5 selainvälimuistia, joka kuitenkin on käytettävyydeltään rajallisempi natiiviapplikaatioihin verrattuna. (Budiou 2013; Marketing & Growth Hacking 2017.)

Webapplikaatiot vaativat vähiten vaivaa asennuksen ja ylläpidon kannalta, eivätkä vie tilaa päätelaiteella. Webapplikaatiota on myös helpoin toteuttaa usealle alustalle yhtäaikaisesti, eivätkä ne vaadi sovelluskauppojen hyväksyntäprosesseja, eikä niiden mahdollisista käyttömaksuista tarvitse antaa osaa sovelluskauppojen ylläpitäjille. (Budiou 2013; Marketing & Growth Hacking 2017.)

Hybridi- ja webapplikaatiot ovat halvempia tuottaa, koska niiden toteutuksessa voidaan käyttää webteknologioita, eikä toteutuksessa tarvita erityistä alustakohtaista ohjelmointikemusta. Alustakohtaista käyttäjäkokemusta mukaileva applikaatio on kuitenkin helpoin toteuttaa natiiviapplikaationa, joka ei kuitenkaan tarkoita, etteikö hybridi- ja webapplikaatiota voisi suunnitella natiivikokemuksen mukaisiksi. (Budiou 2013; Marketing & Growth Hacking 2017.)

### 3 KÄYTTÖLIITYMÄ JA KÄYTTÄJÄKOKEMUS

Käyttöliittymä, ammattikielessä UI (User Interface), terminä sisältää kaiken, millä käyttäjä voi olla vuorovaikutuksessa päätelaitteeseen. Näitä ovat muun muassa näyttö, napaimisto, hiiri ja päätelaiteen näytöllä esitettävät syötteistä ja painikkeista koostuva ulkoasu. (Search Microservices 2016.) Ensimmäisiksi käyttöliittymiksi voidaan lukea erälaskentakoneet, joita käytettiin reikäkorttia tai muuta vastaavaa mediaa hyödyntäen ja joissa oli myös käyttäjäkonsoli. Käyttöliittymät ovat kehittyneet tämän jälkeen komentokehote-käyttöliittymiksi ja graafisiksi käyttöliittymiksi. Kuviossa 3 voidaan nähdä eri tyyppisiä käyttöliittymiä niiden kehityksen varrelta. (Rounds 2016.)



Kuvio 3. Käyttöliittymiä: erälaskentakone, komentokehote-käyttöliittymä ja graafinen käyttöliittymä (Lakeland University 2018; Wikipedia 2018b)

Nykyään käyttöliittymistä puhuttaessa viitataan lähinnä yksinomaan graafisiin käyttöliittymiin. Graafinen käyttöliittymä on sarja näkymiä tai sivuja, jotka koostuvat visuaalisista elementeistä kuten kuvista, painikkeista ja tekstistä. (Lanoue 2016.)

UX, eli käyttäjäkokemus, on käyttäjän omakohtainen kokemus, joka hänellä on tuotetta käytettäessä. UX-termiä käytti ensimmäisenä Don Norman, 1990-luvulla (Lanoue 2016). Norman määritteli esimerkillisen käyttäjäkokemus siten, että sen tulisi täyttää asiakkaan tarpeet yksinkertaisella ja tyylikkäällä tavalla, täsmällisesti ja ilman vaivaa. Käyttäjäkokemuksen tulisi olla enemmän kuin asiakkaan kommunikoimien tarpeiden täyttämistä tai listattujen tuotteen ominaisuuksien tuottamista. Korkealaatuinen käyttäjäkokemus tulisi sisällyttää itseensä saumattomasti kaiken yrityksen tarjonnan, sisältäen tekniikan, markkinoinnin, graafisen- ja teollisen suunnittelun sekä käyttöliittymän. (Nielsen & Norman 2018.)

*Käyttäjäkokemus sisältää kaikki käyttäjän kanssakäymiset yrityksen, sen palveluiden ja sen tuotteiden kanssa.*

- Don Norman (Nielsen & Norman 2018, suomennos kirjoittajan).

On tärkeää erottaa kokonaiskäyttäjäkokemus itse käyttöliittymästä, vaikkakin käyttöliittymä on tärkeä osa käyttäjäkokemusta. Vaikka käyttöliittymä olisikin täydellinen, käyttäjäkokemuksen voi pilata esimerkiksi taustalla toimivan tietokannan sisältämien tietojen puutteellisuus. Samalla tapaa käyttäjäkokemus ja käytettävyys tulisi erottaa toisistaan. Käytettävyys on käyttöliittymän ominaisuus, joka pitää sisällään sen miten helppo järjestelmän käyttö on opetella ja miten tehokasta ja miellyttävää sen käyttö on. Kokonaiskäyttäjäkokeemus on siis laaja termi, joka pitää sisällään enemmän kuin käyttöliittymän tai sen käytettävyyden. (Nielsen & Norman 2018.)

### 3.1 Käytettävyys

Käytettävyys on käyttöliittymän laadukkuuden ominaisuus, jolla määritetään sen käytön helppous. Käytettävyys voidaan määrittää viidellä komponentilla, joihin kuuluvat opittavuus, tehokkuus, muistettavuus, virheellisyys ja tyydyttävyyden. (Nielsen 2012.)

Nämä komponentit vastaavat seuraaviin kysymyksiin:

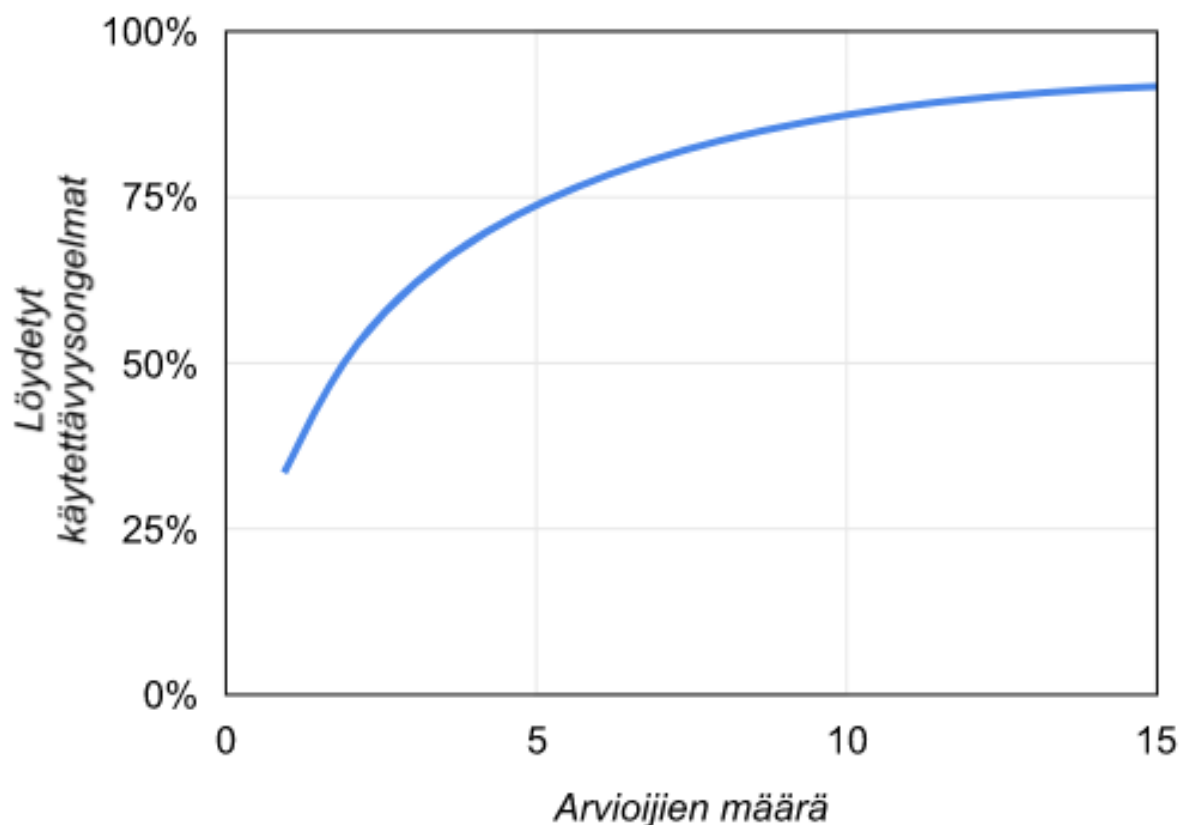
- Opittavuus: miten helppoa käyttäjän on saada ensimmäisellä käyttökerralla haluttu toiminto suoritettua?
- Tehokkuus: miten nopeasti käyttäjä saa toimintoja suoritettua opittuaan käyttöliittymän toiminnallisuuden?
- Muistettavuus: miten helposti käyttäjän on omaksuttava käyttöliittymän toiminnallisuus palatessaan sen käyttöön tauon jälkeen?
- Virheellisyys: miten monta virhettä käyttäjä tekee, miten vakavia nämä virheet ovat ja miten helposti näistä virheistä on toipua?
- Tyydyttävyyden: kuinka mukavaa käyttöliittymän käyttö oli? (Nielsen 2012.)

Toinen tärkeä käyttöliittymän ominaisuus käytettävyyden lisäksi on hyödyllisyys. Hyödyllisyydellä yritetään vastata kysymykseen: ”Tekeekö käyttöliittymä sitä mitä käyttäjä tarvitsee?” Käytettävyys ja hyödyllisyys ovat yhtä tärkeitä käyttöliittymän ominaisuuksia, jotka yhdessä tuottavat, hyvin toteutettuina, käyttökelpoisen käyttöliittymän. (Nielsen 2012.)

### 3.2 Heuristiikka

Käyttöliittymän käytettävyyden arviointia voidaan lähestyä heuristisella arvioinnilla. Heuristinen arviointi on käytettävyystekniikan tapa löytää käytettävyysongelmia käyttöliittymässä

näiden ongelmien iterointia varten. Heuristinen arviointi suoritetaan pienellä joukolla arvioijia, jotka tutkivat käyttöliittymää verraten sen toimintaa ennalta määrättyihin heuristisiin sääntöihin. (Nielsen 1995b.)



Kuvio 4. Heurististen arvioijien vaikutus käytettävyysongelmien löytämiseen (Nielsen 1995b)

Heuristinen arviointi suoritetaan pienessä joukossa, koska yhden ihmisen on vaikea löytää kaikkia käytettävyyteen liittyviä ongelmia yksin ja jokainen yksittäinen arvioija löytää eri ongelmia. Jacob Nielsen onkin omien projektinsa arvioinnin yhteydessä määrittänyt, että yksittäinen arvioija löytää vain noin 35% käytettävyysongelmista. Löydettyjen käytettävyysongelmien määrä kasvaa arvioijien lisääntyessä ylläolevan kuvion 4 tapaan, joka perustuu Nielsenin työstämän kuuden projektin keskiarvoisiin määriin kaavalla 1. Kuvio 4 on vain suuntaa antava esimerkki siitä, miltä kuvaaja yleensä näyttää, kuvaaja vaihtelee siis projekteittain. (Nielsen 1995b.)

$$ProblemsFound(i) = N (1 - (1 - l)^i)$$

$i$  = arvioijien määrä

$N$  = käytettävyyssongelmien määrä käyttöliittymässä

$l$  = yksittäisen arvioijan löytämien käytettävyyssongelmien määrä

Kaava 1. Heuristisen arvioinnin kaava optimaalisen arvioijamäärän selvittämiseen (Nielsen 1995b)

Heuristien arviointi ja käyttäjätestaus ovat eri asioita. Ne eroavat toisistaan siten, että käyttäjätestauksessa tarkkailija toimii arvioijan roolissa, arvioiden käyttäjän toimintoja. Heuristisessa arvioinnissa taas käytettävyyden analysoinnin vastuu on arvioijalla, ja tarkkailijan tehtävä on pitää kirjaa arvioijan antamista kommentteista. Toisin kuin heuristisessa arvioinnissa, jossa tarkkailijan tulisi vastata arvioijan kysymyksiin käyttöliittymän toiminnasta ja käytöstä, käyttäjätestauksessa tarkkailijan ei tulisi puuttua käyttäjän toimiin tai antaa vihjeitä käyttöliittymän käyttöön, koska arvioitava kohde on itse käyttäjä. (Nielsen 1995b.)

Heuristisessa arvioinnissa arviointi suoritetaan jokaisen arvioijan kohdalla eristetyksi. Arviointeja verrataan vasta kun kaikki arviointijoukon jäsenet ovat saaneet arviointinsa päätökseen. Tarkkailija voi auttaa arvioinnin suorittamisessa, jos arviointia ollaan tekemässä epävakaiseen prototyyppiin tai arvioijalle nousee ongelmatilanteita joiden ratkaisuun hänen tekniset taidot eivät riitä. Heuristinen arviointi tapahtuu siten, että arvioija käy käyttöliittymän useaan kertaan läpi, verraten samalla sen käytettävyyttä ennalta määriteltuihin käytettävyyden sääntöihin, eli heuristiikkaan. (Nielsen 1995b.)

Nielsen on määrittänyt vuorovaikutussuunnitteluun kymmenen heuristista "nyrkisääntöä", joita voidaan käyttää käyttöliittymäsuunnittelussa auttamaan käytettävyyden takaamisessa tai arvioidessa käyttöliittymän käytettävyyttä. Näihin sääntöihin lukeutuvat seuraavat asiat (Nielsen 1995a.):

1. Järjestelmän tilan näkyvyys. Järjestelmän tulisi aina pitää käyttäjä tietoisena siitä mitä on tapahtumassa, oikeanlaisen ja oikea-aikaisen palautteen avulla.
2. Järjestelmän ja oikean maailman yhtenevyys. Järjestelmän tulisi puhua käyttäjän kieltä, sanoin ja konseptein, jotka ovat käyttäjälle tuttuja. Tiedon tulisi näyttää luonnolliselta ja sen tulisi olla loogisesti jäsenneltyä.

3. Käyttäjän ohjaus ja vapaus. Käyttäjät valitsevat usein järjestelmästä toimintoja vahingossa ja tarvitsevat nopean pikauloskäynnin epähalutusta järjestelmän tilasta.
4. Johdonmukaisuus ja standardit. Käyttäjän ei tulisi joutua kummastelemaan eri sanojen, tilanteiden tai toimintojen yhtenevyyttä. Tulisi käyttää päätelaitteen parhaita käytäntöjä.
5. Virheiden ehkäisy. Virheilmoitusten sijaan käyttö tulisi suunnitella siten, ettei virheitilanteita tapahtuisi ollenkaan. Virheitä sallivia tiloja tulisi välttää tai tulisi kysyä käyttäjän vahvistusta ennen virheellisen toiminnon suorittamista.
6. Tunnistus muistin sijaan. Toiminnot ja valinnat tulisivat olla näkyvissä siten, ettei käyttäjän tarvitse muistaa niitä. Myös ohjeistus tulisi olla näkyvillä tai helposti saatavilla olevassa paikassa.
7. Joustavuus ja käytön tehokkuus. Järjestelmässä tulisi olla käytettävissä toimintaa nopeuttavia toimintoja kokeneemmille käyttäjille. Käyttäjän tulisi voida räätälöidä käyttöä itselleen sopivaksi.
8. Esteettinen ja minimaalinen muotoilu. Esitettävä informaatio tulisi olla olennaista näkymälle. Ylimääräinen informaatio vähentää olennaisen näkyvyyttä.
9. Käyttäjän auttaminen virheiden tunnistuksessa, diagnosoinnissa ja niistä toipumisessa. Virheilmoitusten tulisi olla helppolukuisia ja täsmällisiä sekä ehdottaa käyttäjälle ratkaisua ongelmaan.
10. Apu ja dokumentointi. Järjestelmää pitäisi pysytä käyttämään ilman dokumentteja. Tarpeen vaatiessa dokumentointiin tulisi kuitenkin tarjota ja siitä tulisi olla helppo etsiä apua, sen tulisi keskittyä käyttäjän toimintoon, listata konkreettisesti vaiheet joita tulisi seurata, eikä se saisi olla liian iso. (Nielsen 1995a.)

Koska heuristinen arviointi liittyy vain käyttöliittymään ja sen käytettävyyteen, eikä sen avulla suoritettavaan toimintoon tai tehtävään, voidaan sitä suorittaa hyvinkin varhaisessa suunnittelun vaiheessa. Arviointia voidaan tehdä esimerkiksi jo silloin kun käyttöliittymäsuunnitelmasta ei ole olemassa kuin paperille piirretty versio. (Nielsen 1995b.)

## 4 KÄYTTÖLIITTYMÄSUUNNITELUN TYÖNKULKU

### 4.1 Rautalankamallinnus

Rautalankamalli voidaan käsittää ulkoasun luurankona. Se mukailee ulkoasun lopullista mallia löyhästi antaen luotettavan kuvan siitä, mihin sisältö asetellaan. (Cao 2018.)

Rautalankamallinnus on helpointa kynällä ja paperilla. Tämän haasteena on kuitenkin paperisten tuotosten vaikea jaettavuus. Koska rautalankamalli on prototyypin esiaste, joudutaan kynästä ja paperista jossain vaiheessa luopumaan. Rautalankamallin paperiversiolla voidaan kuitenkin tehdä alustavaa käytettävyyssprototyyppointia. Näillä paperiprototyypeilla voidaan testata kokeellisia konsepteja nopeasti ja vähin riskein. (Cao 2018.)

Kynän ja paperin käytön lisäksi nykyään on olemassa useita digitaalisia rautalankamallinnustyökaluja. Näillä työkaluilla rautalankamallin tuottaminen on nopeaa, ja useat näistä työkaluista tarjoavat mahdollisuuden mallin muuntamiseen interaktiiviseksi prototyypiksi. Digitaalisten rautalankamallinnustyökalujen käyttö helpottaa suunnitelman jakoa. (Cao 2018.) Rautalankamallinnus voidaan jakaa kahteen työvaiheeseen, joihin kuuluvat applikaation käytön kartoitus ja visuaalinen rautalankamallinnus (Banga & Weinhold 2014, 68 - 69).

#### 4.1.1 Käytön kartoitus

Suunnittelun alussa, rautalankamallinnuksen yhteydessä, on tärkeää miettiä, miten käyttäjä toimii applikaatiossa ensimmäisellä käyttökerralla. On myös ymmärrettävä, miksi käyttäjä on ladannut applikaation, mikä vaikutus applikaatiosta hänellä ensimmäiseksi syntyy ja miten data voidaan esittää hänellä mahdollisimman nopeasti ja tehokkaasti. (Banga & Weinhold 2014, 68.)

Käytön kartoituksen vaiheessa suunnittelua pyritään kartoittamaan ja minimoimaan tarvittavien vaiheiden määrä, jolla käyttäjä pääsee haluttuun lopputulokseen, ja tunnistaa missä käyttöliittymän kohdissa käyttäjällä mahdollisesti on vaikeuksia vuorovaikutuksen kanssa. Kartoitus olisi tärkeä pitää yksinkertaisena ja iteroida nopeasti. Ideaalisesti suunnittelijan tulisi yksinkertaistaa applikaation toimintaa tässä rautalankamallintamisen vaiheessa poistamalla esteitä ja monimutkaisuutta siihen pisteeseen asti, ettei niitä voida enää poistaa, luoden näin optimaalisen käyttökokemuksen. (Banga & Weinhold 2014, 68 - 69.)



#### 4.1.2 Visuaalinen rautalankamallinnus

Visuaalisen rautalankamallinnuksen tarkoitus on mallintaa jokainen käyttöliittymän osa ja vuorovaikutustapa, jota applikaatiossa tullaan käyttämään. Visuaalisessa rautalankamallissa otetaan kantaa myös siihen, tullaanko käyttämään esimerkiksi ääniohjausta tai muita kehittyneempiä vuorovaikutustapoja, joita voi olla käytettävissä. Tämä dokumentaatio on päätapaa, jolla kommunikoidaan applikaation suunniteltua toimintaa muille sen toteuttamiseen osallistuville tahoille, kuten ohjelmoijille. (Banga & Weinhold 2014, 73 - 74.)

Visuaalinen rautalankamalli toimii käyttöliittymän yksinkertaistettuna, matalatarkkuuksisena ulkoasuna elementtihierarkioineen. Tällainen yksinkertaistettu rautalankamallinnus antaa aikaiset suunnitteluvirheet helpommin anteeksi ja antaa aikaa vahvistaa suunnittelun sisällön rakennetta ennen kuin ulkoasuun aletaan lisätä yksityiskohtia. (Cao 2018.)

#### 4.2 Korkeatarkkuuksinen ulkoasumallinnus ja prototypointi

Korkeatarkkuuksinen ulkoasu asettuu rautalankamallin ja prototyypin välimaastoon. Se on staattinen esitys käyttöliittymän visuaalisesta ulkoasusta ja sen toiminnoista (Treder 2018). Visuaalisuutensa takia korkeatarkkuuksisia ulkoasumalleja tuotetaan tyypillisesti piirto-ohjelmilla, kuten Adobe Photoshop tai Illustrator. Useissa prototypointiohjelmissa on kuitenkin työkaluja esimerkiksi vektorigrafiikan tuottamiseen ja näihin ohjelmiin voidaan tuoda käyttöliittymässä käytettäviä graafisia elementtejä tai jopa valmiiksi koottuja päätelaitteen natiivituotteita käyttöliittymäkomponentteja. (Cao 2017; Proto.io 2018.) Korkeatarkkuuksisen ulkoasun tuottaminen ei siis ole sidottu vain piirto-ohjelmiin. Prototypointiohjelman käyttö korkeatarkkuuksisen ulkoasun tuottamiseen helpottaa myös prototypointiin siirtymisessä. (Cao 2017.)

Prototyypin tarkoitus on testata iteroiden tuotetta ennen sen tuottamista, on mahdotonta luoda täydellinen tuote ensikädessä (Hacker 2013, 6; Murphy 2018). Prototypointiprosessin ohittaminen voi johtaa tuotantokustannusten nousuun, näiden virheiden ollessa vaikeammin korjattavissa lopullisessa tuotteessa. Suunnitteluvirheet lopullisessa tuotteessa voivat myös vierastuttaa asiakkaan tai käyttäjän. (Hacker 2013, 6.)

Prototyypin rakentamiseen on olemassa nykyään suurissa määrin ohjelmia, kuten esimerkiksi Adobe XD tai Proto.io-verkkopalvelu, joilla voidaan rakentaa käsin kosketeltavia päätelaitteella toimivia prototyyppejä. Prototypointi voidaan kuitenkin aloittaa jo matalatarkkuuksisella, paperille piirretyllä rautalankamallilla, josta esitys kuviossa 5. Eri tyyppiset prototypointitavat ja -työkalut helpottavat testausta esimerkiksi web- tai muilla lopputeknologioilla toteutettuun prototyyppiin verrattuna. (Murphy 2018.)



Kuvio 5. Paperiprototyyppi (Wikimedia Commons 2018)

Digitaaliset prototyypit toimivat realistisina ja käsin kosketeltavina suunnittelukonseptien todisteina, joilla voidaan testata suunniteltuja käyttöliittymän käytettävyyden olettamuksia, ja niiden avulla voidaan ottaa käyttäjä, asiakas tai muu sidosryhmä helpommin suunnitteluprosessiin mukaan. Prototyyppien avulla voidaan sisällön rakennetta, ulkoasua ja vuorovaikutuskomponentteja testaten edistyä iteroimalla lopullista tuotetta kohti. Prototypointin iterointiprosessi voidaan listata seuraavasti:

1. Käyttäjän tarpeiden tunnistaminen.
2. Ratkaistavan ongelma ja sen ratkaisevan olettamuksen määrittäminen.
3. Prototyypin rakentaminen.
4. Olettamuksen testaus prototyypillä.
5. Prototypointin uudelleen aloittaminen, kunnes lopullinen ratkaisu on valmis. (Murphy 2018.)

Prototypointia tulisi käyttää tuotantoprosessin jokaisessa vaiheessa, aloittaen paperilla ja siirtyen matalatarkkuuksisen rautalankamallin kautta korkeatarkkuusisella visuaalisella suunnittelemalla prototypointiin. Jokainen tuotantoprosessi kuitenkin on erilainen, ja esimerkiksi SaaS-palveluja (Software as a Service) tuottaessa, voi olla tarpeellista lähteä

teknologioiden testausta varten liikkeelle lopputeknologioilla tuotetulla prototyypillä. (Murphy 2018.)

## 5 MATERIAL DESIGN -SUUNNITTELUOHJEISTO

Mobiilikäyttöliittymäsuunnittelussa on usein paras pysyä päätelaitteen alkuperäisen käyttöliittymän suunnittelutavoissa, jolloin on vaikeampi tehdä suunnitteluvirheitä. Alkuperäiset käyttöliittymien suunnittelutavat ovat testattuja, ja ne ovat yleiset sekä usein parhaat tavat toteuttaa toiminnallisuus päätelaitteeseen. Jos suunnitellaan käyttöliittymäkomponentteja tai vuorovaikutustapoja, jotka ovat päätelaitteen käyttöliittymästä poikkeavia, voidaan nopeasti tuottaa käyttäjäkokemus, joka on käyttäjän mukavuusalueen ulkopuolella. (Banga & Weinhold 2014, 66 - 68.)

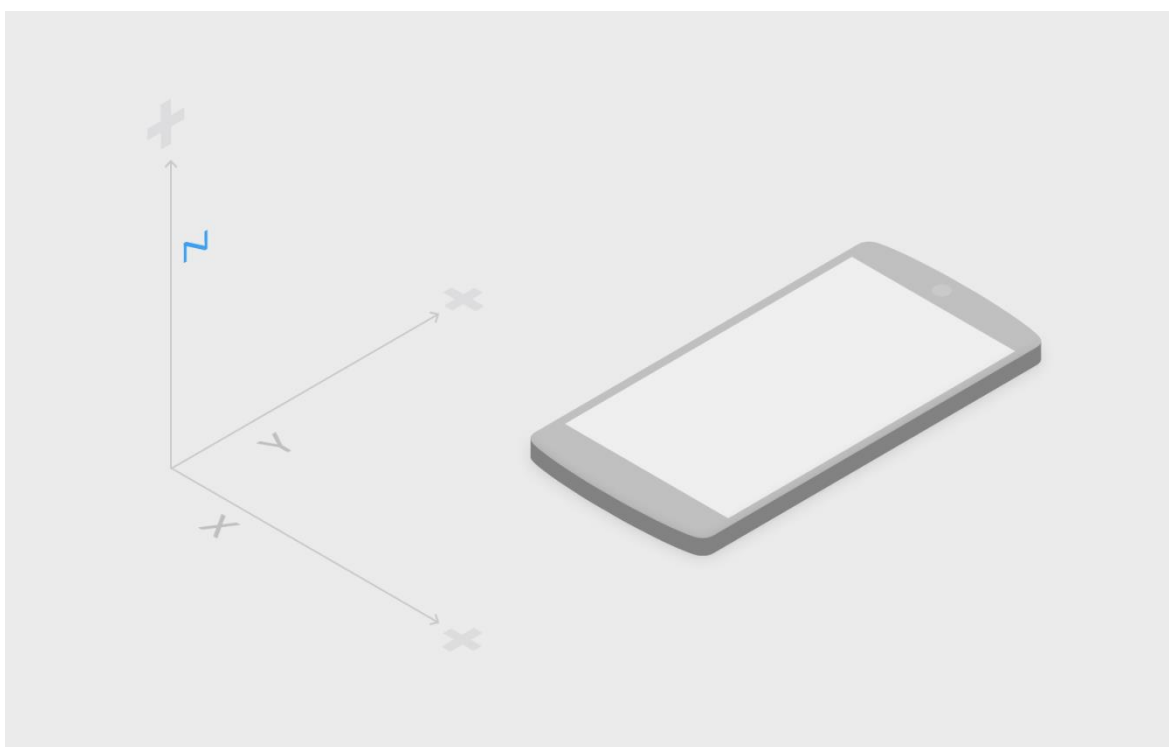
Google on kehittänyt omille palveluilleen ja laitteilleen, kuten Gmail ja Android-käyttöjärjestelmä, Material Design -suunnittelukielen, vuonna 2014 (Simoes 2018). Material Design on skeuomorfismin ja litteän tyylin välimaastossa, esimerkki näistä tyyleistä löytyy kuvio 6, jossa esitetään digitaalista kalenteria skeuomorfisella- ja litteällä tyyllillä suunnitellusti. Litteällä suunnittelutyylillä yritetään minimoida visuaaliset elementit ja tuoda sisältö näin paremmin esille. Material Design yhdistää tämän sisällön edellä viennin ja skeuomorfismin viittaukset fyysisiin kappaleisiin ja lisää näille kappaleille muun muassa massan ja varjot. Nämä fyysiset ominaisuudet vaikuttavat siihen, miten kappaleen tulisi käyttäytyä päätelaitteen ruudulla, ja miten kappale vaikuttaa ympäröivään tilaansa, täten edesauttaen käytön intuitiivisuutta. (Experience 2015.)



Kuvio 6. Skeuomorfisen- ja litteä tyyli (Vyrazu Labs 2016)

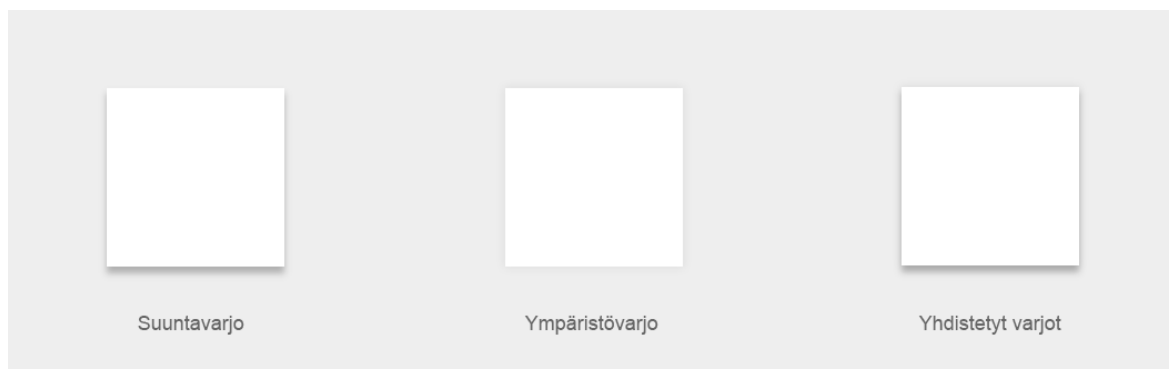
## 5.1 Material-tila

Material Designia ympäröivässä tilassa on kolme ulottuvuutta, eli jokaisella tilassa olevalla kappaleella on mitat ja paikkakoordinaatit x, y, z-koordinaatistossa. Kappaleiden mitat Material-ympäristössä lasketaan dp-yksikköjä käyttäen. Yksi dp-yksikkö vastaa yhden pikselin suuruista aluetta päätelaitteella jonka pikselitarkkuus on 160dpi. Material-tilassa kappaleet ovat aseteltu päätelaitteen pinnalle z-akselin suuntaisesti, ja tätä akselia käytetään kappaleiden järjestelyyn. Kuviossa 7 on visualisoitu Material Design ympäristön koordinaatisto ja päätelaitteen ruudun katselukulma koordinaatistossa. (Google 2018.)



Kuvio 7. Material 3D-tilan koordinaatisto (Google 2018)

Material-kappaleiden z-akselin suuntaista hierarkiaa käyttöliittymässä tuodaan lisäksi esille kappaleiden tuottamilla varjoilla. Material-ympäristössä on yksi suuntavalon joka antaa kappaleille varjon, ja tämä varjo muuttuu kappaleen liikkuesssa z-akselilla. Suuntavalon tuottaman varjon lisäksi kappaleilla on ympäristöstä aiheutuva varjo, joka helpottaa kappaleen muodon havainnointia. Kuviossa 8 on esimerkki 6dp korkeudella olevasta Material-kappaleesta ja Material-ympäristön ja -suuntavalon sille aiheuttamista varjoista. (Google 2018.)

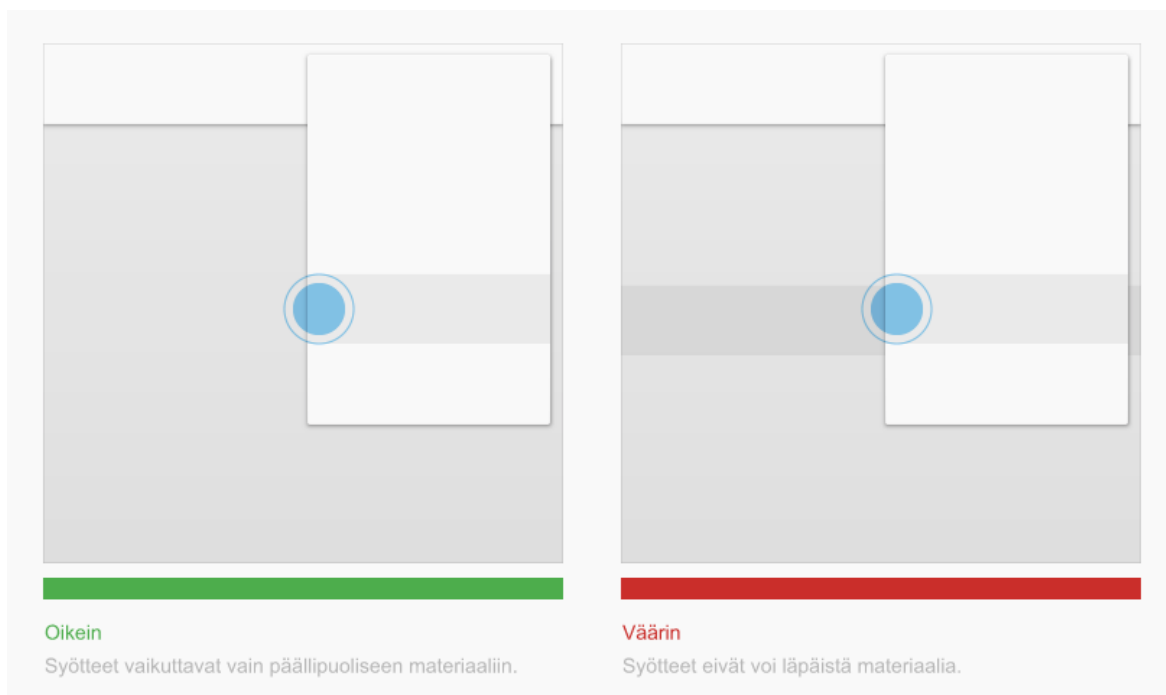


Kuvio 8. Material varjot (Google 2018)

## 5.2 Material-kappaleet

Material-kappaleen mitat x- ja y-akseleilla ovat muuttuvia, mutta niiden paksuus z-akselilla on aina 1dp, eikä kappaleen sisältö vaikuta kappaleen paksuuteen. Kappaleiden sisältö voi olla minkä väristä tai muotoista tahansa ja sisältö voi käyttäytyä kappaleesta itsenäisesti, kuitenkin pysyen kappaleen rajoittaman alueen sisällä. (Google 2018.)

Material-kappaleet ovat kiinteitä, eli ne eivät voi kulkea toistensa läpi, ja esimerkiksi eleohjaukset eivät voi läpäistä kappaletta tai liikkua kappaleen määrittämän alueen ulkopuolelle. Esimerkki kappaleen pinnalla tapahtuvan eleohjauksen oikeellisuudesta kuviossa 9. (Google 2018.)



Kuvio 9. Material-kappaleen eleohjauksen oikeellisuus (Google 2018)

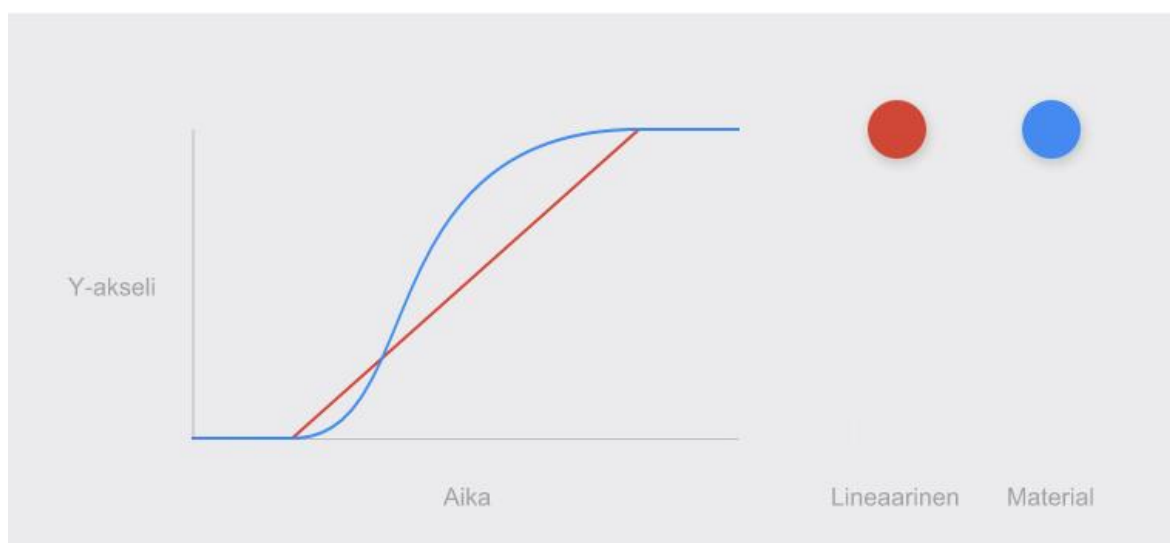
Material-kappaleet voivat vapaasti muuttaa mittojensa lisäksi muotoaan, mutta kappaleen taittaminen ei ole sallittua. Kappaleet voidaan myös luoda tai tuhota spontaanisti, ja vaikka usealla kappaleella ei voi olla päällekkäisyyttä Material-tilassa, voidaan useampi kappale yhdistää yhdeksi kappaleeksi. Sama toimii myös toisin päin, eli kappale voidaan jakaa useaksi erilliseksi kappaleeksi. (Google 2018.)

### 5.3 Material-liike

Material-kappaleiden liikkeiden tarkoitus on ohjata käyttäjää käyttöliittymän navigoinnissa, vihjata hänelle mitä vuorovaikutuksista tapahtuu ja havainnollistaa käyttäjälle käyttöliittymäelementtien hierarkian ja tilan suhteita. Kappaleita voidaan liikuttaa millä akselilla tahansa, ja z-akselin suuntainen liike johtuu tyypillisesti käyttäjän vuorovaikutuksesta. Tällainen vuorovaikutus voi olla esimerkiksi painikkeen painamisesta johtuva painikkeen kohoaminen. (Google 2018.)

Material-liikkeiden kestoon vaikuttaa kappaleen kulkema matka, pitkät animaatioliikkeet ovat tyypillisesti kuitenkin 300-400ms pituudeltaan, ja lyhyet animaatiot 150-200ms. Liikkeen keeseen vaikuttaa kappaleen fysiikka Material-tilassa, jota on imitoitu oikeasta maailmasta. Kappaleen ominaisuudet, kuten paino ja kitka, on otettu huomioon liikkeen tuottamisessa, kuin myös Material-tilassa vallitseva painovoima. Tämä tarkoittaa, että kappale lähtee liikkeelle kiihdyttäen ja liikkeen lopussa pysähtyy paikoilleen hidastuen. Kappaleen

liikettä yhden akselin suuntaisesti liikkeen suorituksen aikana kuvaa kuvio 10, jossa Material-liikettä verrataan lineaariseen liikkeeseen. (Google 2018.)



Kuvio 10. Material-liikkeen kuvaaja verrattuna lineaariseen liikkeeseen (Google 2018)

#### 5.4 Material-työkalut

Itse Material Design -ohjeiston lisäksi sen verkkosivuilta on saatavissa suunnitteluprosessia helpottavia materiaaleja ja työkaluja. Näihin materiaaleihin lukeutuvat eri suunnitteluohjelmille kootut väripaletit ja Material-varjomääritteet sekä Material-komponentteja sisältävät tiedostot. Muuhun suunnittelua nopeuttavaan materiaalin kuuluu muun muassa Material Designia varten valmiiksi tuotetut ikonit ja sen yhteydessä virallisesti käytetyt Roboto ja Noto -fontit. (Google 2018.)

Material Design -työkaluiksi tarjotaan selaimessa toimivia Resizer ja Color -palveluja. Resizer-palvelu tarjoaa mahdollisuuden testata julkaistun sivun responsiivisuutta emuloiduissa selaimessa näytettävässä laiteruudussa. Color-palvelussa voidaan tarkastella ennalta määritettyjä Material-väripaletin tai käyttäjän itse määrittämiä värejä erilaisissa valmiissa käyttöliittymäkompositioissa. (Google 2018.)



## 6 CASE: SALPAUSSELKÄ GEOPARK -MOBIILIAPPLIKAATIO

### 6.1 Lähtötilanne

Salpausselät ovat ainutlaatuinen ja geologisesti kansainvälisesti merkittävä muodostuma, joka koostuu kolmesta reunamuodostumasta (Wikipedia 2018a). Salpausselät ja niiden syntyhistoria on vaikuttanut niiden kattaman alueen luonnon erityispiirteisiin, ilmastoon ja elinkeinoelämän kehittymiseen, mutta niitä ei ole vielä täysimääräisesti hyödynnetty luontomatkailun puolesta. (Räsänen 2018.)

Salpausselkä Geopark on Lahden ammattikorkeakoulun hallinnoima hanke, joka toteutetaan yhteistyössä Metsähallituksen, Geologian tutkimuskeskuksen sekä Salpausselkien läheisyyteen sijoittuvien kuntien kanssa, joihin voidaan lukea mukaan Asikkala, Heinola, Hollola, Kärkölä, Lahti, Padasjoki ja Sysmä. Hanke on aloitettu vuonna 2017, ja se tulee kestämaan vuoden 2019 loppuun. Hankkeen tavoitteena on lisätä Salpausselkä UNESCO Global Geopark -verkostoon. (Räsänen 2018.)

UNESCO Global Geoparkit ovat yksittäisiä yhtenäisiä maantieteellisiä alueita, jotka koostuvat kansainvälisesti geologisesti merkittävistä kohteista. Geoparkeja on tarkoitus hallita suojelulla, koulutuksella ja kestäväällä kehityksellä. Tällä hetkellä Geopark-verkostoon kuuluu maailmanlaajuisesti 140 Geoparkia, 38 eri maassa. (UNESCO 2018.)

Yhtenä Salpausselkä Geopark -hankkeeseen liittyvänä tavoitteena on tuottaa Salpausselkä Geoparkille mobiiliapplikaatio. Tätä palvelua oli alustavasti suunniteltu verkkosivuina toimivaksi karttapalveluksi ennen kuin asiakas oli päättänyt uudelleenohjaavansa palvelun mobiiliapplikaatioksi.

Mobiiliapplikaation tulisi tarjota palvelu jolla käyttäjä voisi tutustua Geoparkin alueeseen, sen sisältämiin kohteisiin ja reitteihin. Lisäksi käyttäjän tulisi pystyä applikaatiota apuna käyttäen liikkumaan Geopark-alueella. Applikaation tulisi siis toimittaa reaaliaikainen paikannus ja karttapohja.

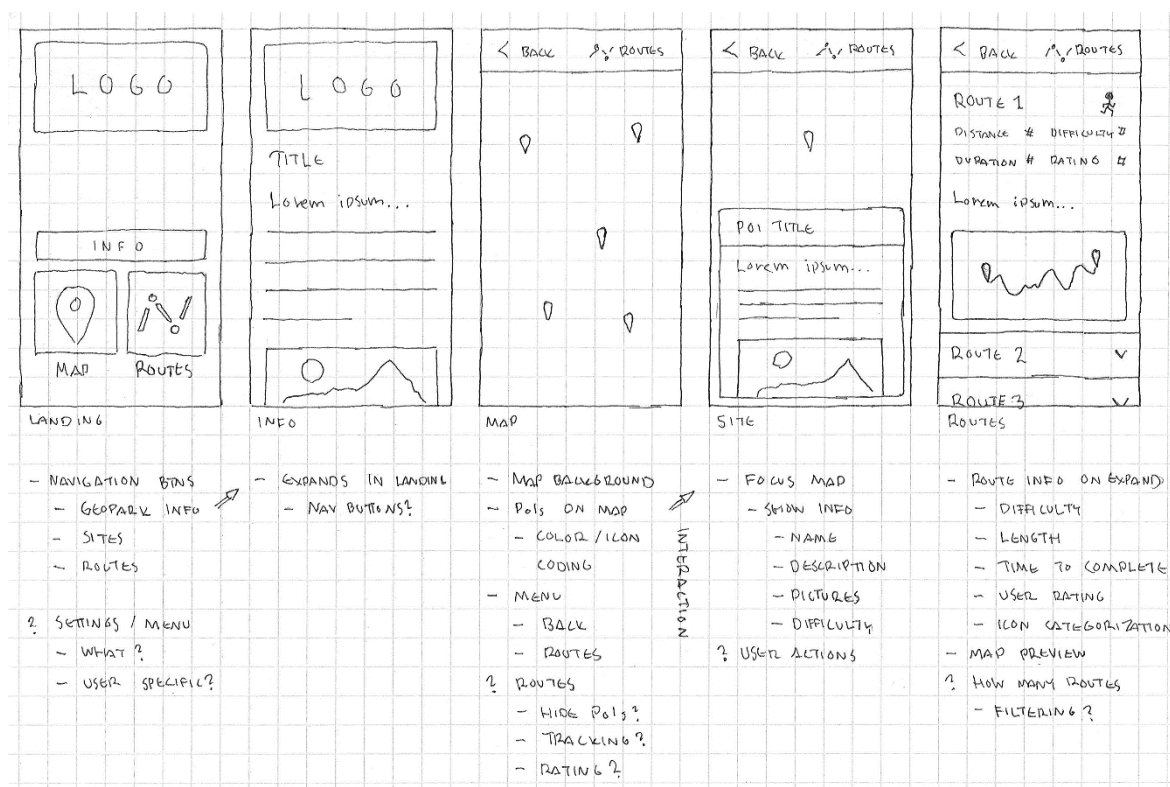
Mobiiliapplikaatio tulotaisiin toteuttamaan hybridiapplikaationa webteknologioita käyttäen, jotta se olisi mahdollista tarpeen vaatiessa helposti muokattavissa myös työpöytäselaimessa toimivaksi. Koska applikaation tulisi skaalautua siis erisuuruisten mobiililaitteiden lisäksi mahdollisesti työpöytäkäyttöön, lähdettiin sen suunnittelua toteuttamaan mobile-first -käytännöin.

Ennen suunnittelun aloittamista, tutustuttiin kilpaileviin tuotteisiin, joihin lukeutuivat Magma Geopark- ja Rokua Geopark -mobiiliapplikaatiot. Nämä applikaatiot tuotiin esille asiakkaan

toimesta, koska ne sisälsivät mahdollisesti haluttuja toiminnollisuuksia tai tarjosivat helpommin ymmärrettävissä olevan esityksen joillekin asiakkaan toiveille applikaatioon halutuista toiminnoista. Asiakkaan kanssa käydyissä palavereissa kiinnitettiin myös huomiota eri sisältöjen kategorisointiin LahtiMob-applikaatiossa, ja Magma Geopark -applikaation kohde ja reittierottelu sekä karttaesitys tuotiin myös esille.

## 6.2 Rautalankamallin toteutus

Kilpailevien tuotteiden analysoinnin jälkeen aloitettiin matalatarkkuuksisen rautalankamallin tuottaminen. Rautalankamallinnuksella kuvattiin applikaation eri näkymät, jotka jaoteltiin kuvion 11 tavalla aloitus- ja infosivuun sekä kartta-, kohde- ja reittinäkymiin. Näille näkymille määritettiin kunkin sisältämät toiminnot, sekä miten suunnitellut näkymät liittyivät toisiinsa näillä toiminnoilla.

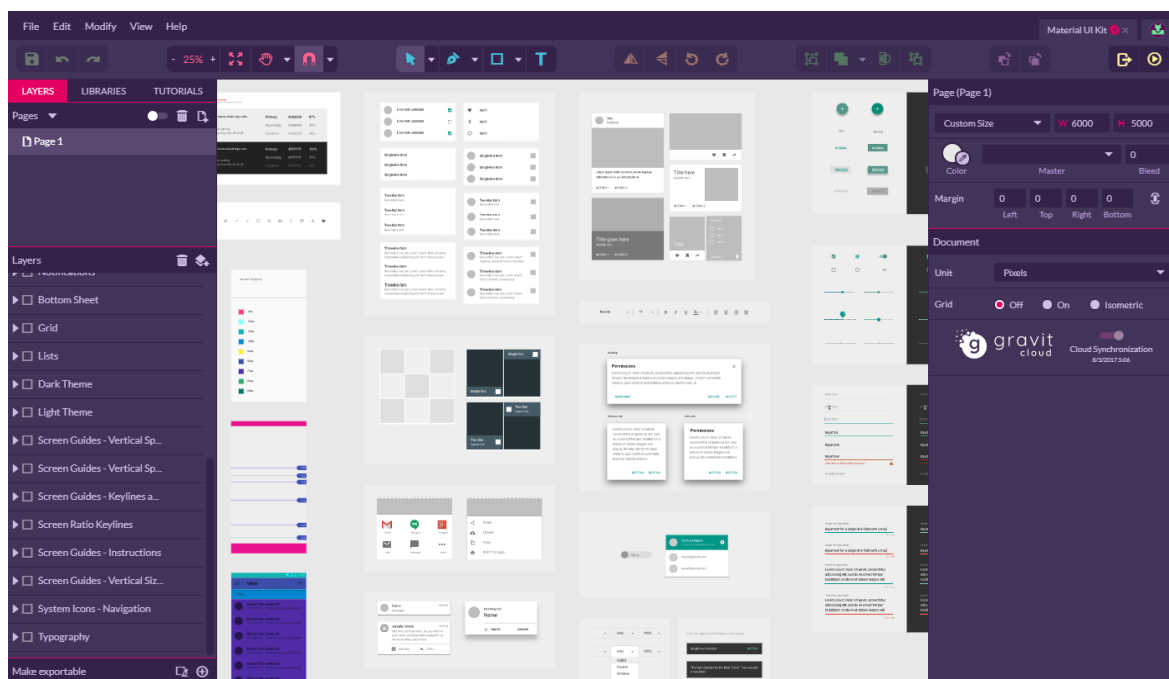


Kuvio 11. Matalatarkkuuksinen rautalankamalli Salpausselkä Geopark -mobiliapplikaation käyttöliittymästä ja sen käytön kulusta

Matalarkkuuksisessa rautalankamallissa ei kiinnitetty varsinaisesti huomiota eri käyttöliittymäelementtien asetteluun, vaan niiden määrittely jätettiin korkeatarkkuuksisen ulkoasu-suunnittelun toteuttamiseen, jolloin valittiin myös applikaatiossa käytettävä Material Design -tyyliohjeisto ohjaamaan ulkoasun rakennetta.

### 6.3 Korkeatarkkuuksinen ulkoasu

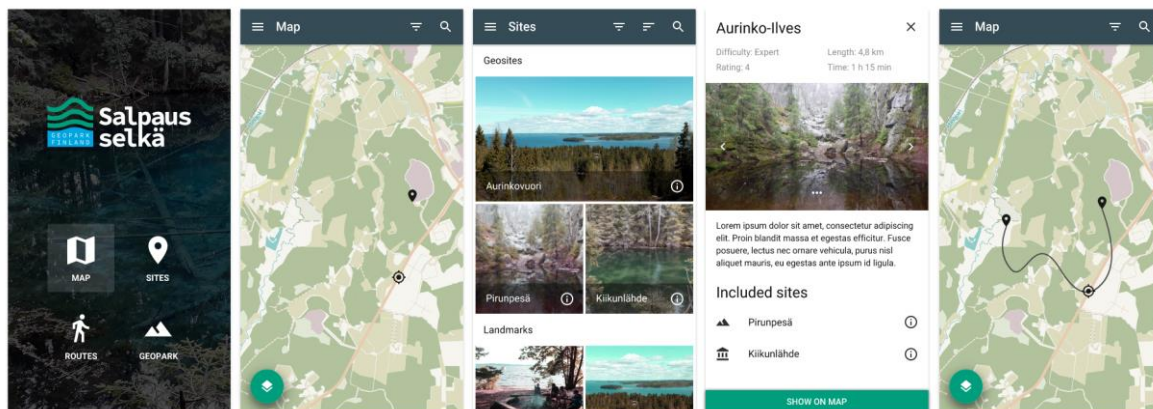
Matalatarkkuuksisen rautalankamallin pohjalta aloitettiin korkeatarkkuuksisen ulkoasun suunnittelu Gravit Designer -ohjelmalla, joka on ilmainen vektoripohjainen suunnitteluohjelma. Ulkoasun suunnittelun yhteydessä otettiin käyttöön Material Design -suunnitteluohjeisto. Material Design -ohjeiston käyttöä suunnitteluohjelmassa helpotti sille toteutettu Material-komponenttikirjasto, josta havainne kuviossa 12. Ohjeisto ei kuitenkaan kiinnitä huomiota kaiken tyyppisen sisällön jäsentelemiseen, joten suunnittelun aikana oli ajoittain tarve soveltaa ohjeiston antamaa tyyliisuuntaa sisällön rakenteen suunnittelussa.



Kuvio 12. Valmiit Material-komponentit Gravit Designer -ohjelmassa (Dragos 2018)

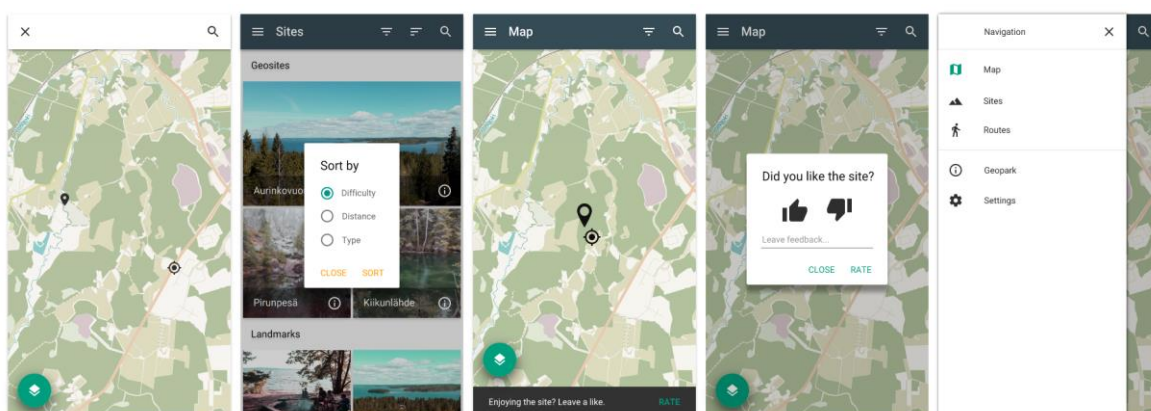
Koska Material Design -ohjeistolla oli suuri vaikutus siihen, miten käyttöliittymän toiminnot ja sisältö tultiin asettelemaan, tehtiin suurin osa käyttöliittymän rakenteen iteroinnista korkeatarkkuuksisen ulkoasun kautta, suunnitteluohjelmaa käyttäen. Iteroinnin aikana matalatarkkuuksisessa rautalankamallissa suunnitellut näkymät uudelleen määrittyivät kuvion

13 tapaan, jossa esitetään aloitus-, kartta-, kohdelista- ja reittivalintanäkymät sekä reitin esitys karttanäkymässä.



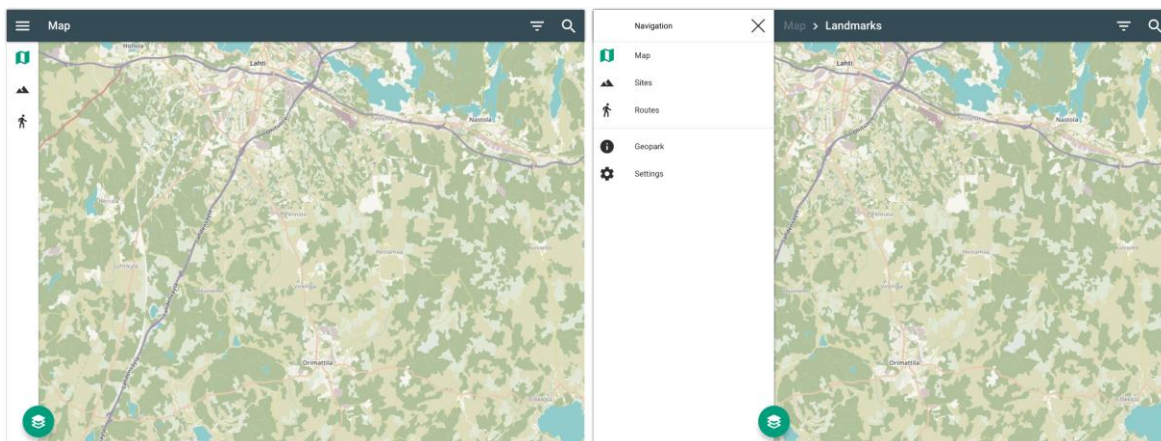
Kuvio 13. Näkymien uudelleen määrittäykset korkeatarkkuuksisessa ulkoasusuunnitelmassa

Korketarkkuuksisen ulkoasusuunnitelman iteroinnin edetessä otettiin enenevissä määrin kantaa eri näkymissä käyttäjälle tarjottaviin toimintoihin, joita kuviossa 11 esitetyssä matalatarkkuuksisessa rautalankamallissa oli suunniteltu, kuten sisällön etsiminen, lajittelu ja suodatus sekä palautteen kysely reittiä kulkiessa tai kohteella käydessä. Näiden toimintojen iteroinnissa käytettiin apuna ulkoasusuunnitelman eri versioiden pohjalta tehtyjä prototyyppjejä, ja asiakkaan antamaa palautetta. Eri näkymien sisältämien toimintojen lisäksi navigaatio eri näkymien välillä mietittiin uudelleen. Esitys näistä toiminnoista ja muutetusta navigaatiosta on kuviossa 14.



Kuvio 14. Näkymien toimintoja ja navigaatio applikaatiossa

Kun älypuhelin kokoinen ulkoasu oli saatu suunniteltua useimpien näkymien osalta, lähdettiin ulkoasusuunnitelmaa mukauttamaan myös tablettikokoon. Koska käytetyt Material Designin määrittämät komponentit olivat skaalautuvia, voitiin näyttöalueen laajetessa keskittyä lähinnä vain sisällön uudelleenjärjestelyyn ja toimintojen sekä näytettävän tiedon lisäämiseen. Esimerkki tablettikoon vaikutuksesta ulkoasuun voidaan nähdä kuviossa 15, jossa vasemmalla nähdään navigaation olevan esillä jatkuvasti, antaen käyttäjälle mahdollisuuden navigaatioikoneihin tottuessa käyttää niitä päänäkymien välillä navigointiin. Kuviossa 15 oikealla taas nähdään navigaatio suurennettuna, jolloin käyttäjälle tarjotaan selitteet navigointikomponenteille sekä lisävaihtoehtoja navigointiin applikaation sisällä. Samassa kuviossa on myös esimerkki siitä, miten käyttäjälle voidaan antaa lisätietoa ruudun koon kasvaessa. Kuviossa näkymän otsikkoon on lisätty käyttäjän tekemän suodatuksen tyyppi.



Kuvio 15. Applikaation navigaatio tablettikoossa ja lisätietoa suodatuksesta

## 6.4 Prototypointi ja käytettävyytestaus

Korkeatarkkuuksisen käyttöliittymäsuunnitelman pohjalta suoritettiin prototypointia ja käytettävyytestausta. Prototyypeillä pyrittiin aluksi HTML-, CSS- ja JavaScript -teknologioiden avulla määrittämään käyttöliittymän toteutusvaiheeseen käytettävissä olevaa applikaation HTML-rakennetta. Lisäksi näiden prototyyppien pohjalta voitiin suorittaa käytettävyytestausta, jonka avulla käyttöliittymäsuunnitelman iterointia voitiin jatkaa.

Käyttöliittymäsuunnitelman laajentuessa webteknologioiden käyttäminen ei enää kuitenkaan ollut ajankäytön kannalta soveltuva lähestymistapa prototypointiin. Siksi projektin edetessä otettiin käyttöön Proto.io-palvelu. Proto.io on selaimessa toimiva applikaatio,

jolla voidaan toteuttaa käyttöliittymäsuunnittelua ja prototypointia laitekoosta riippumatta (kuvio 16).

## New Project

Create a new blank prototype

PROJECT NAME

Geopark

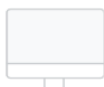
PROJECT TYPE



SMART PHONE



TABLET



WEB  
DESKTOP



SMART WATCH  
INTERNET OF THINGS



CUSTOM

DEVICE SKIN / SCREEN SIZE

Android Mobile (360 x 640)

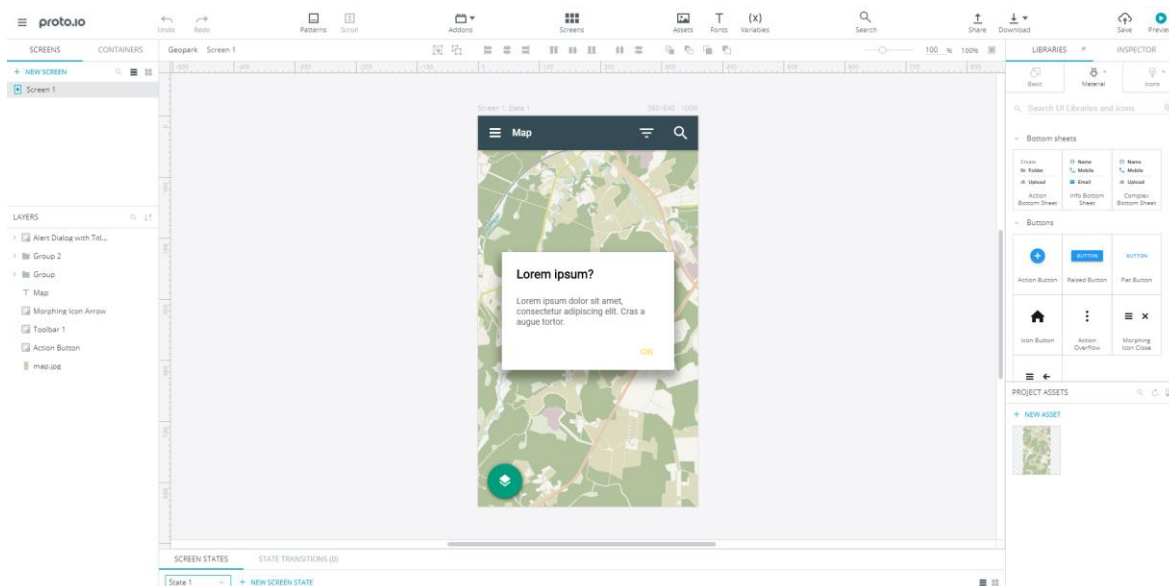
CANCEL

CREATE

Kuvio 16. Proto.io-palvelun projektin luonti (Proto.io 2018)

Palveluun voitiin tuoda korkeatarkkuuksisen käyttöliittymäsuunnitelman näkymiä ja lisätä näihin näkymiin interaktiivisuutta vuorovaikutusalueita käyttäen. Palvelulla oli myös mahdollista luoda käyttöliittymäsuunnitelmanäkymiä kokonaisuudessaan, ja palvelu tarjosi esimerkiksi Android-alustan natiiviyylisiä Material Design -elementtejä, kuvion 17 tapaan, käytettäväksi suunnittelussa, joten Android-ympäristön natiiviuuden toteutusta suunnittelussa ei ollut tarpeen tehdä käsin.





Kuvio 17. Proto.io-palvelun käyttöliittymä ja Material Design -komponentit (Proto.io 2018)

Palvelun tuottamaan prototyyppiin voitiin toteuttaa käyttöliittymän toiminnallisuutta lisäämällä kuviossa 18 esitetyllä tavalla toimintoja prototyyppiin rakennetuille komponenteille tai tuodun käyttöliittymäsuunnitelman päälle piirretyille vuorovaikutusalueille. Eri näkymien välille voitiin esimerkiksi luoda animoitu siirtymä tällä tavoin. Yksittäisille näkymille oli mahdollista asettaa myös eri tiloja, jos prototypissä haluttiin esittää tai testata jotain sen näkymän sisäistä toimintoa.

Go to screen...

Title

Interaction 1

Trigger

Tap

Action

Go to screen...

Screen

Delay

0

ms

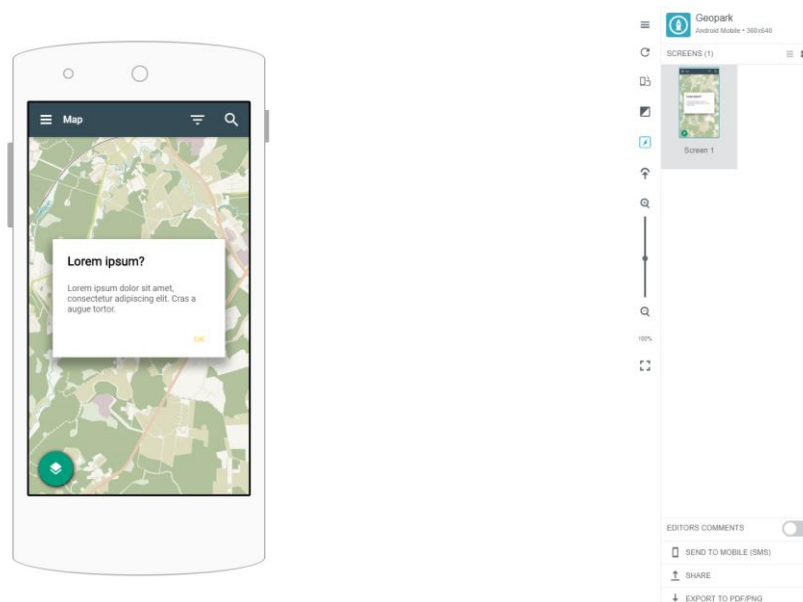
☐ Callback  
 Executed after the action is completed

CANCEL

SAVE INTERACTION

Kuvio 18. Siirtymän lisäys käyttöliittymäkomponenttiin Proto.io-palvelussa (Proto.io 2018)

Kun palvelulla tuotettu prototyyppi oli valmis, tai jos sen toimintaa haluttiin testata prototyyppin työstövaiheessa, voitiin prototyyppi emuloida palvelun avulla selaimessa kuviossa 19 esitetyllä tavalla. Tämä emuloitu prototyyppi oli myös ladattavissa HTML-muodossa palvelun ulkopuolella käytettäväksi.



Kuvio 19. Proto.io-palvelulla emuloitu mobiilikäyttöliittymä selaimessa (Proto.io 2018)



## 7 YHTEENVETO

Salpausselkä Geopark -applikaatiolle saatiin tuotettua Material Design -ohjeistoa noudattava älypuhelinkokoinen ulkoasusuunnitelma kokonaisuudessaan. Tätä suunnitelmaa voidaan jatkossa hyödyntää applikaation ulkoasun toteuttamisessa. Ulkoasusuunnitelmalla otettiin myös kantaa siihen, miten sen tulisi skaalautua tabletin ruudulle, ja millä tavalla näytettävän sisällön ja tiedon tulisi mukautua käytettävän tilan kasvaessa.

Salpausselkä Geopark -projektin ollessa kuitenkin vielä meneillään vuoden 2019 loppuun asti, jäi ulkoasun suunnittelemisen osittain kesken. Suunnitelmissa ei otettu kantaa karttopohjalla esitettäviin kohteita tai reittejä kuvastavaan ikonografiaan tai siihen millä tavalla Material Design -animaatiot tulisi toteuttaa näkymien välillä, animaatioiden osalta applikaation kehitystä jatkava taho voi kuitenkin tukeutua Material Design -ohjeistoon.

Koska applikaatio tullaan toteuttamaan webteknologoin, voidaan osa käyttöliittymän vuorovaikutusanimaatioista toteuttaa Material Designia toteuttavia tyylikirjastoja käyttäen. Ulkoasusuunnitelmat tuskin tulevat olemaan myöskään viimeisin versio toteutettavasta ulkoasusta, koska kehityksen yhteydessä tehtävä prototypointi voi hyvinkin tuoda esiin käytettävyyssongelmia, joiden takia applikaation ulkoasua tai toimintoja tulee olla tarpeen muuttaa.

## LÄHTEET

Banga, C. & Weinhold, J. 2014. Essential Mobile Interaction Design: Perfecting Interface Design in Mobile Apps. Indiana: Addison-Wesley.

Bar, A. 2018. What Web Can Do Today [viitattu 25.4.2018]. Saatavissa: <https://what-webcando.today/>

Budiu, R. 2013. Mobile: Native Apps, Web Apps, and Hybrid Apps [viitattu 27.2.2018]. Saatavissa: <https://www.nngroup.com/articles/mobile-native-apps/>

Cao, J. 2017. All you need to know about mockups, wireframes, and prototypes [viitattu 4.5.2018]. Saatavissa: <https://www.creativebloq.com/web-design/jargon-wireframes-mockups-prototypes-51514898/2>

Cao, J. 2018. What is a Wireframe: Designing Your UX Backbone [viitattu 10.4.2018]. Saatavissa: <https://www.uxpin.com/studio/ui-design/what-is-a-wireframe-designing-your-ux-backbone/>

Dragos, C. 2018. GitHub - CristianDragos/gravit-material-ui-kit: Material Design sticker sheet for Gravit Designer [viitattu 4.5.2018]. Saatavissa: <https://github.com/CristianDragos/gravit-material-ui-kit>

Experience, A. 2015. THE USABILITY OF VISUAL DESIGN FRAMEWORKS AND GOOGLE'S MATERIAL DESIGN [viitattu 2.5.2018]. Saatavissa: <http://uxbert.com/the-usability-of-visual-design-frameworks-and-googles-material-design/>

Google 2018. Material Design [viitattu 13.4.2018]. Saatavissa: <https://material.io/guidelines>

Gremillion, B. 2018. A Hands-On Guide to Mobile-First Responsive Design [viitattu 23.2.2018]. Saatavissa: <https://www.uxpin.com/studio/blog/a-hands-on-guide-to-mobile-first-design/>

Hacker, W. 2013. Mobile Prototyping With Axure 7. Birmingham: Packt Publishing.

Kharlampidi, V. 2016. Framework7 [viitattu 27.2.2018]. Saatavissa: <https://framework7.io/>

Lakeland University 2018. Computer Interface [viitattu 10.4.2018]. Saatavissa: <http://cps.luj.tokyo/sw04.php>

Lanoue, S. 2016. UI vs. UX: What's the difference between user interface and user experience? [viitattu 10.4.2018]. Saatavissa: <https://www.usertesting.com/blog/2016/04/27/ui-vs-ux/>

Marketing & Growth Hacking 2017. Native vs. Web vs. Hybrid Apps - What's the Difference? [viitattu 27.2.2018]. Saatavissa: <https://blog.markgrowth.com/native-vs-web-vs-hybrid-apps-whats-the-difference-1df4c5e4bc50>

Materialize 2018. About – Materialize [viitattu 27.2.2018]. Saatavissa: <http://materializecss.com/>

Murphy, C. 2018. A Comprehensive Guide to Wireframing and Prototyping [viitattu 4.5.2018]. Saatavissa: <https://www.smashingmagazine.com/2018/03/guide-wireframing-prototyping/>

Nielsen, J. 1995a. 10 Usability Heuristics for User Interface Design [viitattu 25.4.2018]. Saatavissa: <https://www.nngroup.com/articles/ten-usability-heuristics/>

Nielsen, J. 1995b. How to Conduct a Heuristic Evaluation [viitattu 25.4.2018]. Saatavissa: <https://www.nngroup.com/articles/how-to-conduct-a-heuristic-evaluation/>

Nielsen, J. 2012. Usability 101: Introduction to Usability [viitattu 25.4.2018]. Saatavissa: <https://www.nngroup.com/articles/usability-101-introduction-to-usability/>

Nielsen, J. & Norman, D. 2018. The Definition of User Experience (UX) [viitattu 10.4.2018]. Saatavissa: <https://www.nngroup.com/articles/definition-user-experience/>

Proto.io 2018. Our latest features [viitattu 30.4.2018]. Saatavissa: <https://proto.io/>

Rounds, D. 2016. A Short History of Computer User Interface Design [viitattu 10.4.2018]. Saatavissa: <http://blog.usabilla.com/short-history-computer-user-interface-design/>

Räsänen, P. 2018. Salpausselästä UNESCO:n geopuistokohde [viitattu 30.4.2018] Saatavissa: [http://www.theseus.fi/bitstream/handle/10024/142303/LAMK\\_2018\\_36.pdf?sequence=1&isAllowed=y](http://www.theseus.fi/bitstream/handle/10024/142303/LAMK_2018_36.pdf?sequence=1&isAllowed=y)

Search Microservices 2016. User interface [viitattu 10.4.2018]. Saatavissa: <https://searchmicroservices.techtarget.com/definition/user-interface-UI>

Simoos, A. 2018. Google's Material Design - Android Design Language [viitattu 14.4.2018]. Saatavissa: <https://www.interaction-design.org/literature/article/google-s-material-design-android-design-language>

Statcounter 2018. Desktop vs Mobile vs Tablet Market Share Worldwide [viitattu 22.5.2018]. Saatavissa: <http://gs.statcounter.com/platform-market-share/desktop-mobile-tablet>

Treder, M. 2016. Wireframing, Prototyping, Mockuping – What's the Difference? [viitattu 4.5.2018]. Saatavissa: <https://designmodo.com/wireframing-prototyping-mockuping/>

UNESCO 2018. UNESCO Global Geoparks [viitattu 30.4.2018]. Saatavissa: <http://www.unesco.org/new/en/natural-sciences/environment/earth-sciences/unesco-global-geoparks/>

Vyrazu Labs 2016. Skeuomorphic design vs Flat design: which design technique to use, and why? [viitattu 2.5.2018]. Saatavissa: <http://www.vyrazu.com/blog/design-technique-to-use/>

W3C 2015. Graceful degradation versus progressive enhancement [viitattu 28.2.2018]. Saatavissa: [https://www.w3.org/wiki/Graceful\\_degradation\\_versus\\_progressive\\_enhancement](https://www.w3.org/wiki/Graceful_degradation_versus_progressive_enhancement)

Wikimedia Commons 2018. File:ELiving Campus Paper Prototype 2.jpg [viitattu 4.5.2018]. Saatavissa: [https://commons.wikimedia.org/wiki/File:ELiving\\_Campus\\_Paper\\_Prototype\\_2.jpg](https://commons.wikimedia.org/wiki/File:ELiving_Campus_Paper_Prototype_2.jpg)

Wikipedia 2018a. Salpausselät [viitattu 30.4.2018]. Saatavissa: <https://fi.wikipedia.org/wiki/Salpausselät>

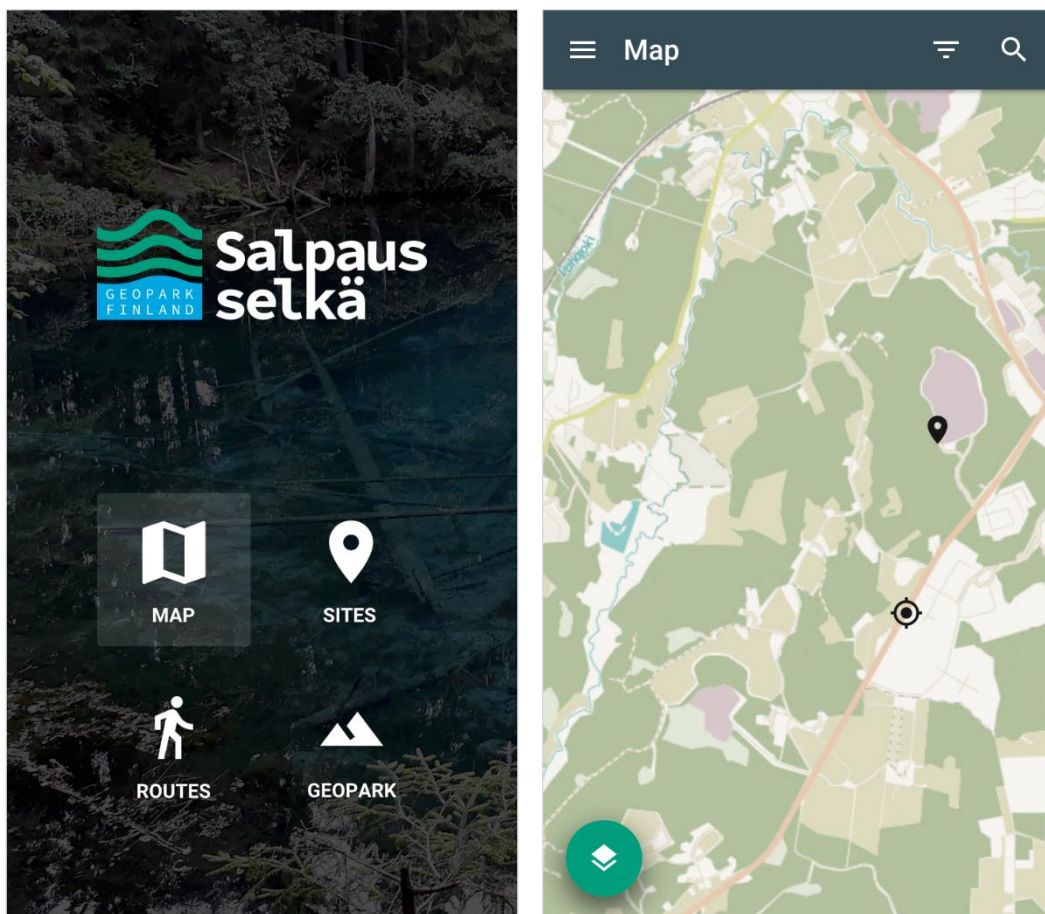
Wikipedia 2018b. User Interface [viitattu 10.4.2018]. Saatavissa: [https://en.wikipedia.org/wiki/User\\_interface](https://en.wikipedia.org/wiki/User_interface)

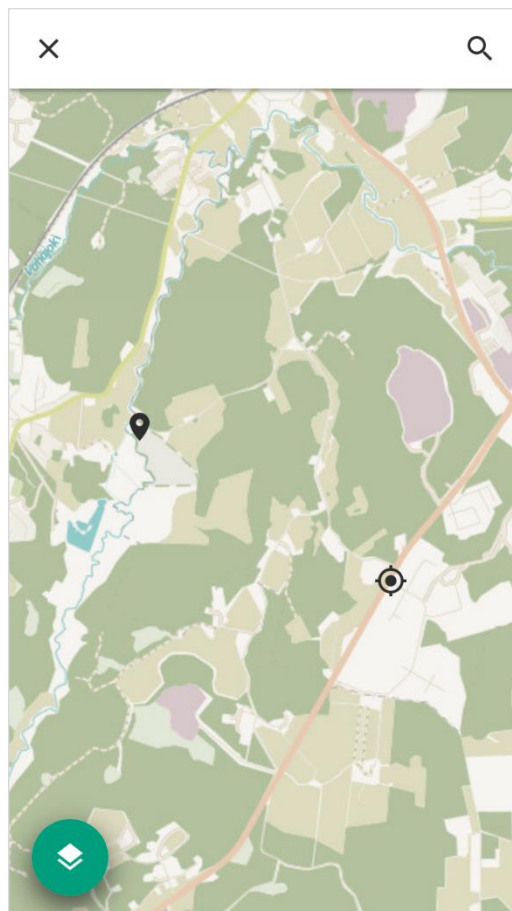
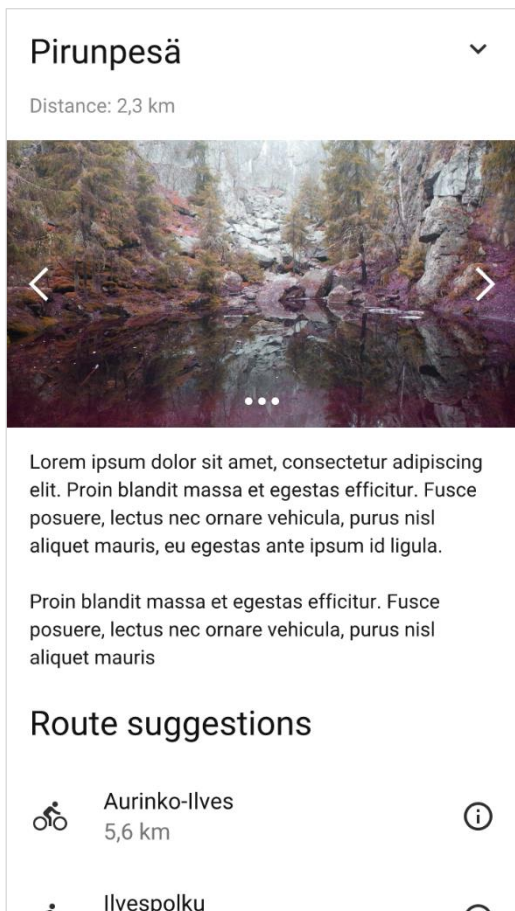
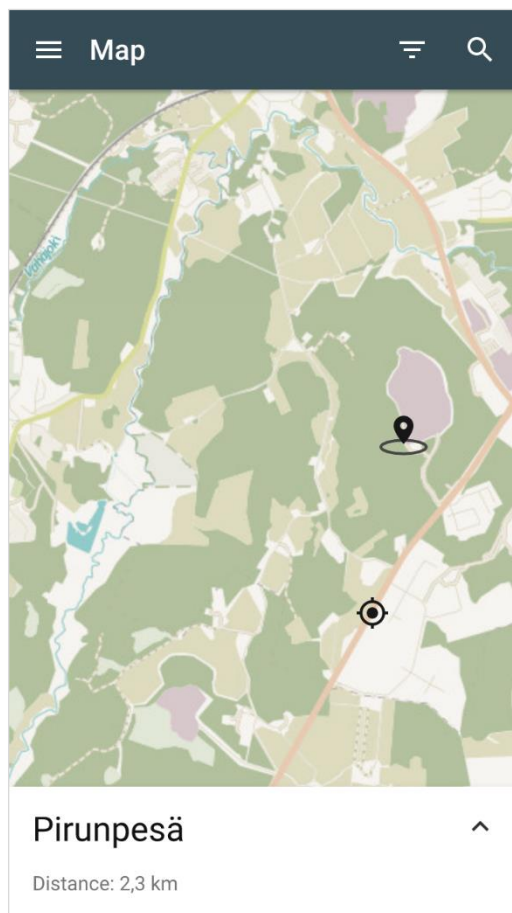
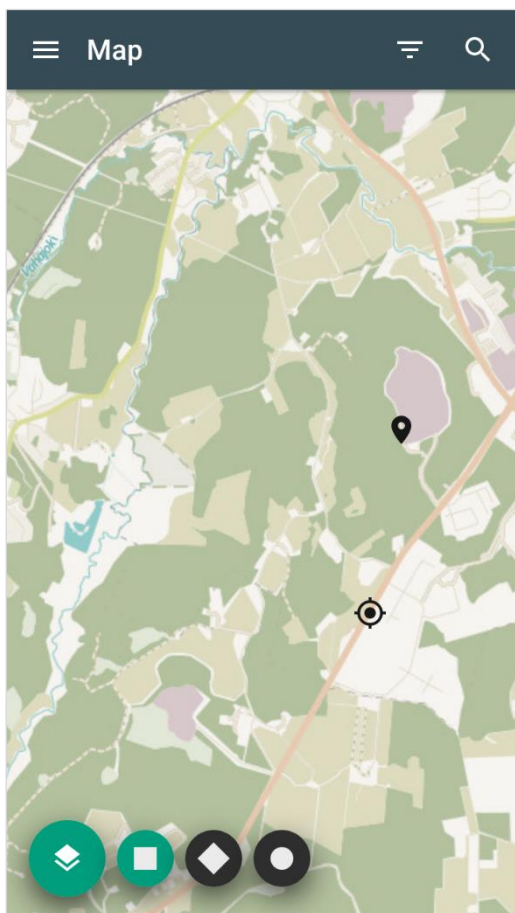
Wroblewski, B. 2009. Mobile First [viitattu 23.2.2018]. Saatavissa: <https://www.lukew.com/ff/entry.asp?933>

Wroblewski, L. 2010. Mobile First Helps with Big Issues [viitattu 23.2.2018]. Saatavissa: <https://www.lukew.com/ff/entry.asp?1117>

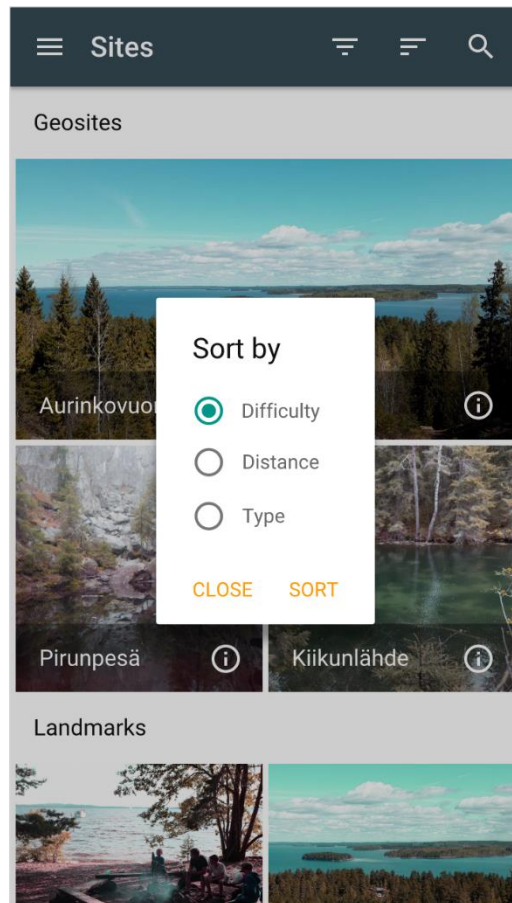
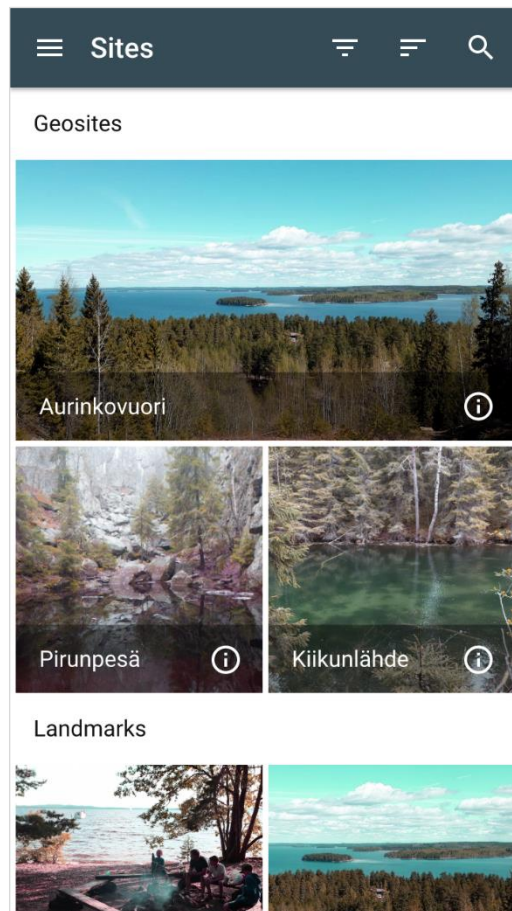
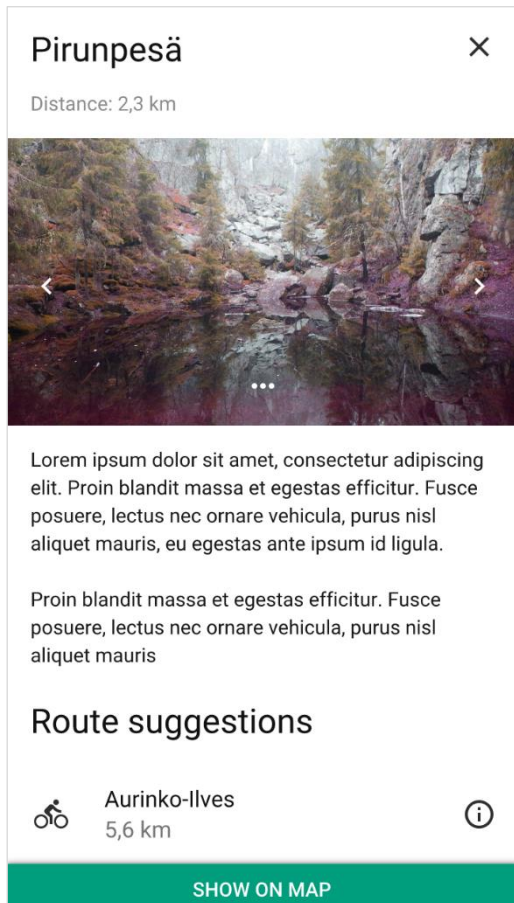
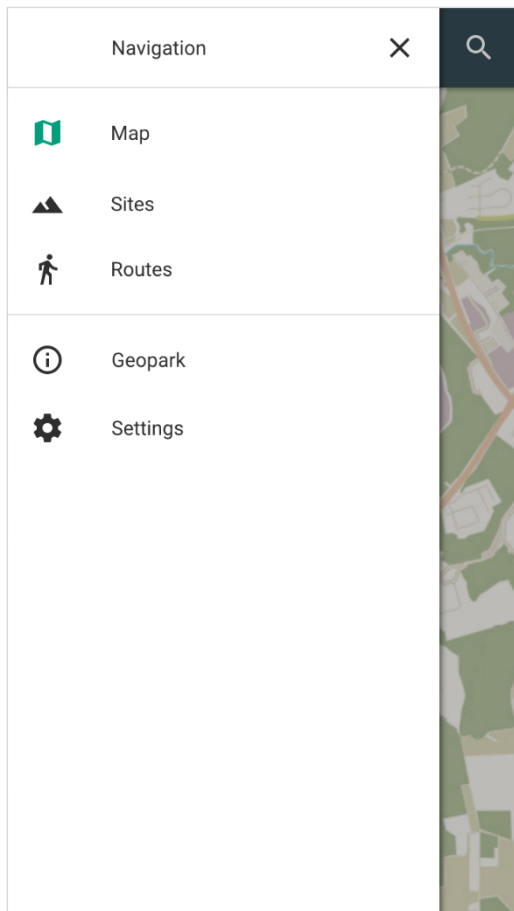
## LIITTEET

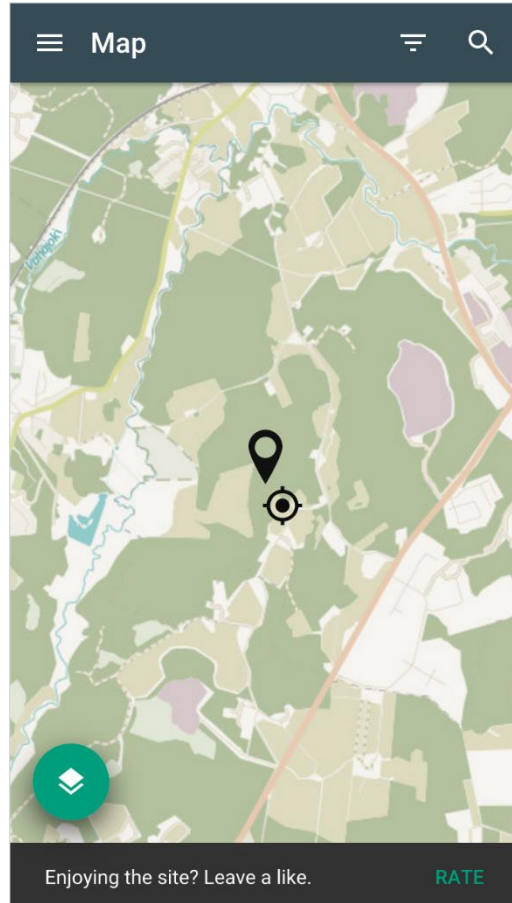
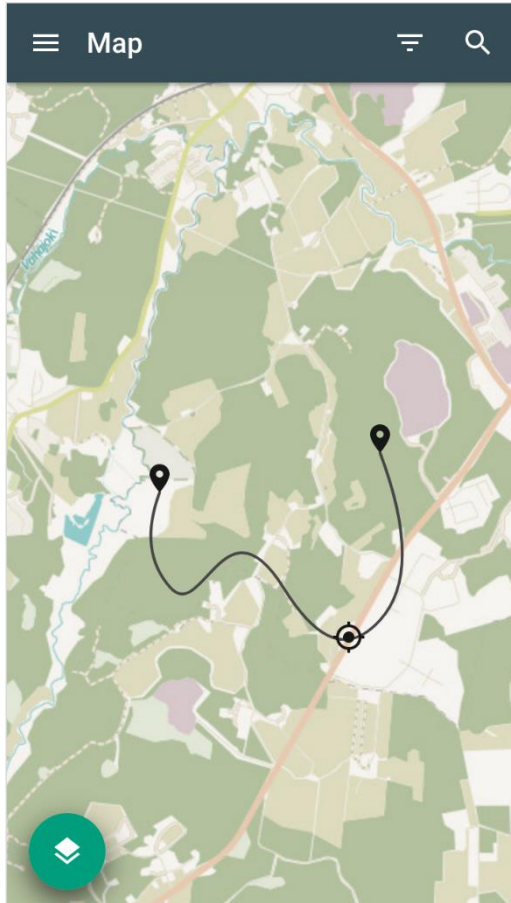
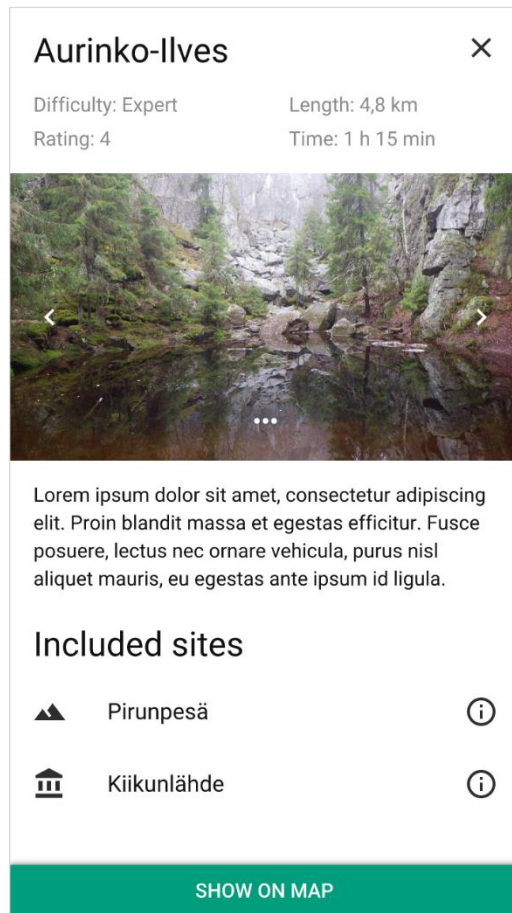
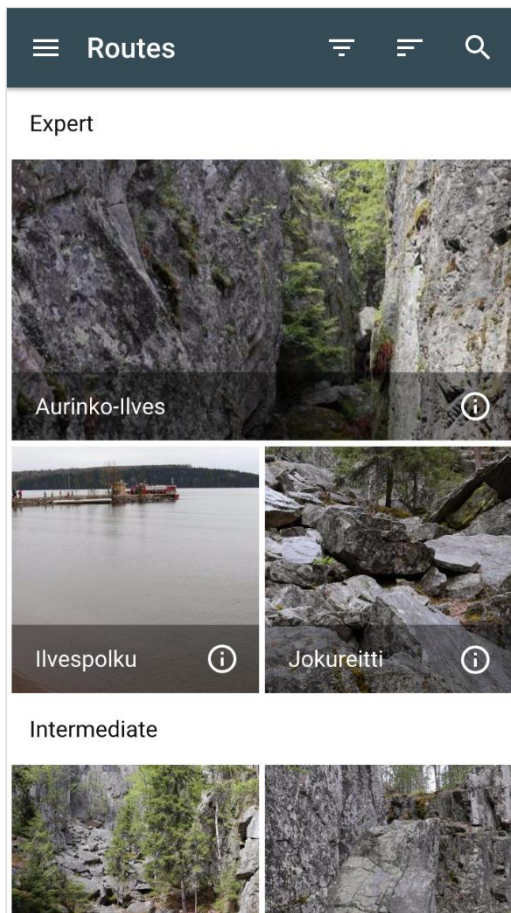
Liite 1 Mobiilikäyttöliittymän korkeatarkkuuksinen suunnitelma



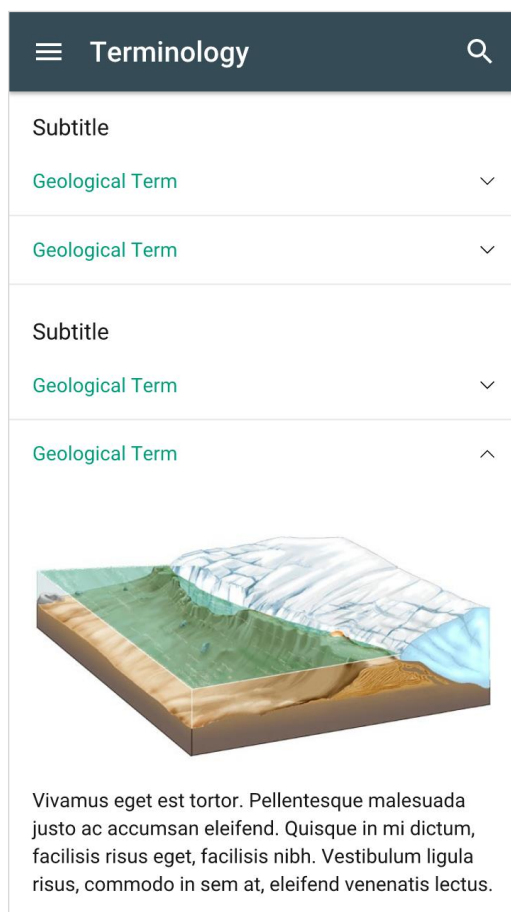
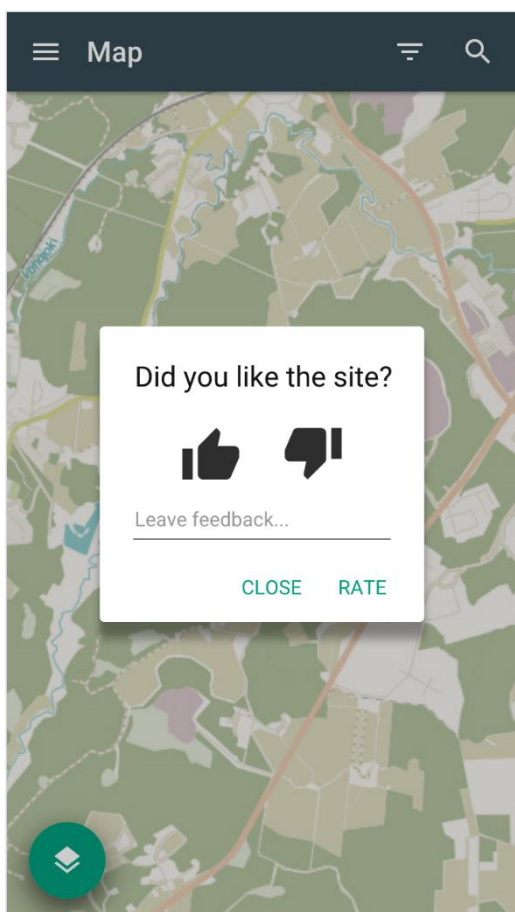












## Liite 2 Tablettikäyttöliittymän korkeatarkkuuksinen suunnitelma

